

**SGT University, Chandu-Budhera, Gurugram**  
**Faculty of Engineering & Technology**  
**Department of Computer Science & Engineering**



**Masters of Computer Application**  
**Scheme & Syllabus (2021-22 Onwards)**

**Vision of SGT University**

**“Driven by Research & Innovation, we aspire to be  
amongst the top ten Universities in the Country by 2022”**

# Master of Computer Application

## Semester 1st

S. No.	Subject Code	Subject Name	L	T	P	C	Internal	External	Total
1		Problem Solving and Programming with C++(OOPS)	3	0	0	3	40	60	100
2		Software Testing	3	0	0	3	40	60	100
3		Advanced Database Management System	3	0	0	3	40	60	100
4		Department Electives-1	3	0	0	3	40	60	100
5		Networking technologies	3	0	0	3	40	60	100
6		Problem Solving and Programming with C++(OOPS) Lab	0	0	2	1	60	40	100
7		Software Testing Lab	0	0	2	1	60	40	100
8		Networking technologies Lab	0	0	2	1	60	40	100
9		Advanced DataBase Management System Lab	0	0	2	1	60	40	100
10		Department Electives-1 Lab	0	0	2	1	60	40	100
11		Value Addition Course-I	2	0	0	2	40	60	100
		<b>Total</b>	<b>17</b>	<b>0</b>	<b>10</b>	<b>22</b>			

# Master of Computer Application

## Semester 2nd

S. No.	Subject Code	Subject Name	L	T	P	C	Internal	External	Total
1		Computer Organization & Architecture	3	0	0	3	40	60	100
2		Advanced JAVA	3	0	0	3	40	60	100
3		OE-I	4	0	0	4	40	60	100
4		Medical image processing	3	0	0	3	40	60	100
5		Advanced Operating System	3	0	0	3	40	60	100
6		Data Structures & Algorithm Design	3	0	0	3	40	60	100
7		Advanced Operating System Lab	0	0	2	1	60	40	100
8		Advanced JAVA Lab	0	0	2	1	60	40	100
9		Data Structures & Algorithm Design Lab	0	0	2	1	60	40	100
		<b>Total</b>	<b>19</b>	<b>0</b>	<b>6</b>	<b>22</b>			

# Master of Computer Application

## Semester 3rd

S. No.	Subject Code	Subject Name	L	T	P	C	Internal	External	Total
1		Programming in Python	2	0	0	2	40	60	100
2		Research Methodology	3	0	0	3	40	60	100
3		Department Electives-II	3	0	0	3	40	60	100
4		OE-II	4	0	0	4	40	60	100
5		Mobile App Development	3	0	0	3	40	60	100
6		Programming in Python Lab	0	0	4	2	60	40	100
7		Department Electives-II Lab	0	0	2	1	60	40	100
8		Mobile App Development Lab	0	0	2	1	60	40	100
9		Value Addition Course-II	2	0	0	2	40	60	100
		<b>Total</b>	<b>17</b>	<b>0</b>	<b>8</b>	<b>21</b>			

# Master of Computer Application

## Semester 4th(A)

S. No.	Subject Code	Subject Name	L	T	P	C	Internal	External	Total
1		Department Electives-III	3	0	0	3	40	60	100
2		Embedded systems in Medicine	3	0	0	3	40	60	100
3		Department Electives-III Lab	0	0	2	1	60	40	100
4		Dissertation	0	0	4	2	100	100	200
		<b>Total</b>	<b>6</b>	<b>0</b>	<b>8</b>	<b>10</b>			

# Master of Computer Application

## Semester 4th(B)

S. No.	Subject Code	Subject Name	L	T	P	C	Internal	External	Total
1		Internship	0	0	0	10	100	100	200
		<b>Total</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>10</b>			

# MCA

## Semester I

### Problem Solving and Programming with C++(OOPS)

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	<b>Problem Solving and Programming with C++(OOPS)</b>	<b>L</b>	<b>T</b>	<b>P</b>		
<b>3. Course Code</b>		<b>3</b>	<b>0</b>	<b>0</b>		
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE()</b>		<b>OE ()</b>	
<b>5. Pre-requisite (if any)</b>	Programming in 'C'	<b>6. Frequency (use tick marks)</b>	Even	Odd (✓)	Either Sem ()	Every Sem ()
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 0</b>		
<b>8. Course Description</b>						
<p>This course introduces the concepts of object-oriented programming to students with a background in the procedural paradigm. The course begins with a brief review of control structures and data types with emphasis on structured data types and array processing. It then moves on to introduce the object-oriented programming paradigm, focusing on the definition and use of classes along with the fundamentals of object-oriented design. Other topics include an overview of programming language principles, simple analysis of algorithms, basic searching and sorting techniques, event-driven programming, memory management and an introduction to software engineering issues.</p>						
<b>9. Learning objectives:</b>						
<ol style="list-style-type: none"> <li>1. To develop programming skills of students, using object oriented programming concepts.</li> <li>2. To learn the concept of class and object using C++ and develop classes for simple applications.</li> <li>3. To Identify importance of object oriented programming and difference between structured oriented and object oriented programming features.</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
1. Implement, test and debug solutions in C++.						
2. Identify the relative merits of different algorithmic designs.						
3. Apply good programming style and understand the impact of style on developing and maintaining programs.						
4. Design object oriented solutions for small systems involving multiple objects.						
<b>11. Unit wise detailed content</b>						
<b>Unit-1</b>	<b>Number of lectures = 9</b>					
<p><b>Problem Solving Concepts:</b> General Problem Solving Concepts-Types of problems, problems solving with computers, difficulties with problem solving, Problem Solving Aspects, Problem Solving Concepts for computer- constants and variables, data types, functions, operators, expressions and equations.</p>						

<b>Unit – 2</b>	<b>Number of lectures = 9</b>	
<p><b>Foundations of Object Oriented Programming :</b>Introduction to procedural, modular, object-oriented and generic programming techniques, Limitations of procedural programming, Need of object-oriented programming, fundamentals of object-oriented programming: objects, classes, data members, methods, messages, data encapsulation, data abstraction and information hiding, inheritance, polymorphism.</p> <p><b>Extensions to C :</b> Variable declarations, global scope, ‘const’, reference variables, operators in C++(scope resolution, new , delete), dynamic memory allocation, function prototypes, default and constant arguments, ‘cin’, ‘cout’, inline functions</p> <p><b>Class:</b> Defining a class, data members and member functions, public, private and protected members, inline member functions, static data members, static member functions, constructors, destructors, array of objects, classes, objects and memory, class as ADTs and code reuse</p>		
<b>Unit – 3</b>	<b>Number of lectures = 09</b>	
<p><b>Overloading and Inheritance:</b> Function overloading, friend function, friend class Operator Overloading : Introduction, Need of operator overloading, rules for operator overloading, overloading the unary and binary operators using member function, operator overloading using friend function, overloading relational and logical operators, overloading new, delete and assignment operator, type conversions Inheritance : Introduction, Need of inheritance, base and derived classes, member access control, types of inheritance, derived class constructor, constructors in multiple inheritance, overriding member functions, ambiguity in multiple inheritance, virtual base class Virtual functions : Pointers to objects, ‘this’ pointer, Pointers to derived class, virtual function, rules for virtual function, pure virtual function, abstract class, virtual destructors, early and late binding, container classes,</p>		
<b>Unit – 4</b>	<b>Number of lectures = 9</b>	
<p><b>Templates, Exception Handling and File I/O:</b> Namespaces: Introduction, Rules of namespaces, Templates : Introduction, Function template and class template, overloading function template, member function templates and template arguments Exception Handling: Introduction, Exception handling mechanism: try, catch and throw, Multiple Exceptions, Exceptions with arguments Managing Console I/O Operations: Introduction, C++ streams, stream classes, unformatted I/O, formatted I/O and I/O manipulators File I/O: Introduction, Classes for file stream operations, file operations (open, close, read, write, detect end of file), file modes, File pointers and their manipulations, error handling during file operations</p>		
<p><b>12. Brief Description of self-learning / E-learning component</b>  The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University.  The link to the E-Learning portal.  <a href="https://elearning.sgtuniversity.ac.in/course-category/OOPS">https://elearning.sgtuniversity.ac.in/course-category/OOPS</a></p>		
<b>13. Books Recommended</b>		
<b>Text Books</b>		
<ul style="list-style-type: none"> <li>• Programming and Problem Solving with C++ By Nell B. Dale, Chip Weems, 6th edition Jones &amp; Bartlett Publishers, 2014</li> </ul>		
<b>Reference Books</b>		
<ul style="list-style-type: none"> <li>• "Problem Solving with C++ " by Walter Savitch Sixth Edition Pearson/Addison-Wesley, 2007</li> <li>• Programming with C++ by John R. Hubbard, Atul Kahate, 3rd Edition,schaums series 2009</li> </ul>		

## Semester I

### SOFTWARE TESTING

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	<b>SOFTWARE TESTING</b>	<b>L</b>	<b>T</b>	<b>P</b>		
<b>3. Course Code</b>		<b>3</b>	<b>0</b>	<b>0</b>		
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE()</b>		<b>OE ()</b>	
<b>5. Pre-requisite (if any)</b>	Software Engineering	<b>6. Frequency (use tick marks)</b>	Even	Odd (✓)	Either Sem ()	Every Sem ()
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 0</b>		
<b>8. Course Description</b>						
<p>This course will examine fundamental software testing and related program analysis techniques. Emerging concepts such as test-case prioritization and their impact on testing will be examined. Students will gain hands-on testing/analysis experience via a multi-phase course project.</p>						
<b>9. Learning objectives:</b>						
<ol style="list-style-type: none"> <li>1. Appreciate the fundamentals of software testing and its application through the software life cycle.</li> <li>2. Develop skills in designing and executing software tests suitable for different stages in the software life cycle.</li> <li>3. Understand and appreciate the role of software testing in systems development, deployment and maintenance.</li> <li>4. Develop a continuing interest in software testing, and obtain satisfaction from its study and practice.</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
<ol style="list-style-type: none"> <li>1. Acquire knowledge of basic principles and knowledge of software testing and Debugging and test cases.</li> <li>2. Understand the perceptions on testing like levels of testing, generalized pseudo code and with related examples</li> <li>3. Study the various types of testing.</li> <li>4. Analyze the difference between functional testing and structural testing.</li> </ol>						
<b>11. Unit wise detailed content</b>						
<b>Unit-1</b>	<b>Number of lectures = 9</b>					
<p><b>Introduction:</b> What is software testing and why it is so hard?, Some Software Failures, Error, Fault, Failure, Incident, Test Cases, Testing Process, Limitations of Testing, V Shaped Software Life Cycle Model, No absolute proof of correctness, Overview of Graph Theory.</p> <p><b>Verification Testing:</b> Verification Methods, SRS Verification, Software Design Document Verification, Code Reviews, User Documentation Verification, Software Project Audits.</p> <p><b>Functional Testing:</b> Boundary Value Analysis, Equivalence Class Testing, Decision Table Based Testing, Cause Effect Graphing Technique.</p> <p><b>Structural Testing:</b> Identification of Independent Paths: Control Flow Graph, DD-Paths, Cyclomatic Complexity, Graph Matrix, Control Flow Testing, Data Flow Testing, Slice Based Testing, Mutation testing.</p>						



<b>Unit – 2</b>	<b>Number of lectures = 9</b>	
<p><b>Use Case Testing:</b> Use Case Diagrams and Use Cases, Generation of Test Cases from Use Cases, Applicability. Validity Checks: Strategy for Data Validity, Guidelines for Generating Validity Checks. Database testing. Selection, Minimization</p> <p><b>Prioritization of test cases for Regression Testing:</b> Regression Testing, Regression Test Case Selection, Prioritization guidelines, Priority category Scheme, Code Coverage Techniques for Prioritization of Test Cases, Risk Analysis.</p>		
<b>Unit – 3</b>	<b>Number of lectures = 09</b>	
<p><b>Testing Activities:</b> Unit Testing, Levels of Testing, Integration Testing, System Testing, Debugging Object Oriented Testing: Issues in Object Oriented Testing, Path testing, Class Testing, state based testing, Object Oriented Integration and System Testing.</p> <p><b>Metrics and Models in Software Testing:</b> What are Software Metrics, categories of Metrics, object Oriented Metrics used in testing, What should we measure during testing?, Software Quality Attributes.</p> <p><b>Prediction Model:</b> Reliability Modes, Fault Prediction Model.</p>		
<b>Unit – 4</b>	<b>Number of lectures = 9</b>	
<p><b>Test Management and Automation</b> Test Planning, Management, Execution and Reporting, Software Test Automation: Scope of automation, Design &amp; Architecture for automation, Generic requirements for test tool framework, Test tool selection, Testing in Object Oriented Systems.</p>		
<p><b>12. Brief Description of self-learning / E-learning component</b> The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University.</p>		
<p><b>13. Books Recommended</b></p>		
<p><b>Text Books</b></p> <ul style="list-style-type: none"> <li>• Effective Methods for Software Testing, William Perry, John Wiley &amp; Sons, New York, Van Nostrand Reinhold, New York, 2nd Ed., 2006.</li> </ul>		
<p><b>Reference Books</b></p> <ul style="list-style-type: none"> <li>• Software Testing A Craftsman’s approach, Paul C. Jorgenson, CRC Press.</li> <li>• Testing Computer Software, Cem Kaner, Jack Falk, Nguyen Quoc, Van Nostrand Reinhold, New York, 2nd Ed.</li> <li>• Foundation of Software Testing, Aditya P. Mathur, Pearson, 2008.</li> <li>• Software Engineering – A Practitioner’s Approach, Roger S. Pressman, McGraw-Hill International Edition, New Delhi, 5th Ed..</li> <li>• Software Engineering, K. K. Aggarwal &amp; Yogesh Singh, New Age International Publishers, New Delhi, 3rd Ed.</li> </ul>		

## Semester I

### Advanced Database management System

Name of the Department- Computer Science & Engineering						
<b>Course Name</b>	<b>Advanced Database management System</b>	<b>L</b>	<b>T</b>	<b>P</b>		
<b>Course Code</b>		<b>3</b>	<b>0</b>	<b>0</b>		
<b>Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE()</b>		<b>OE ()</b>	
<b>Prerequisite (if any)</b>	<b>DBMS</b>	<b>Frequency (use tick marks)</b>	<b>Even ()</b>	<b>Odd (✓)</b>	<b>Either Sem ()</b>	<b>Every Sem ()</b>
<b>Total Number of Lectures, Tutorials, Practical (assuming 12weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 0</b>		
<b>Course Description</b>						
<p>This module aims to give students in depth information about system implementation techniques, data storage, representing data elements, database system architecture, the system catalog, query processing and optimization, transaction processing concepts, concurrency control techniques, database recovery techniques.</p>						
<b>Learning objectives:</b>						
<ol style="list-style-type: none"> <li>1. To understand the basic concepts and terminology related to DBMS and Relational Database Design</li> <li>2. To the design and implement Distributed Databases.</li> <li>3. To understand advanced DBMS techniques to construct tables and write effective queries, forms, and reports</li> </ol>						
<b>Course Outcomes (COs):</b>						
<ol style="list-style-type: none"> <li>1. Exposure for students to write complex queries including full outer joins, self-join, sub queries, and set theoretic queries.</li> <li>2. Know how of the file organization, Query Optimization, Transaction management, and database administration techniques</li> </ol>						
<b>Unit wise detailed content</b>						
<b>Unit-1</b>	<b>Number of lectures = 9</b>					
<p>Formal review of relational database and FDs Implication, Closure, its correctness 3NF and BCNF, Decomposition and synthesis approaches, Basics of query processing, external sorting, file scans.</p>						
<b>Unit – 2</b>	<b>Number of lectures = 9</b>					
<p>Processing of joins, materialized vs. pipelined processing, query transformation rules, DB transactions, ACID properties, interleaved executions, schedules, serializability</p>						
<b>Unit – 3</b>	<b>Number of lectures = 9</b>					
<p>Correctness of interleaved execution, Locking and management of locks, 2PL, deadlocks, multiple level granularity, CC on B+ trees, Optimistic CC</p>						
<b>Unit – 4</b>	<b>Number of lectures = 9</b>					
<p>Time stamped, lock based techniques, Multiversion approaches, Comparison of CC methods, dynamic databases, Failure classification, recovery algorithm, XML and relational databases</p>						

**Brief Description of self-learning / E-learning component**

The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University.

The link to the E-Learning portal.

<https://elearning.sgtuniversity.ac.in/course-category/>

**Books Recommended**

## Text Books

R. Ramakrishnan, J. Gehrke, Database Management Systems, McGraw Hill, 2004

A. Silberschatz, H. Korth, S. Sudarshan, Database system concepts, 5/e, McGraw Hill, 2008.

## Reference Books

K. V. Iyer, Lecture notes available as PDF file for classroom use.

## Semester I

### Networking technologies

<b>1. Name of the Department: CSE</b>						
<b>2. Course Name</b>	<b>Networking technologies</b>	<b>L</b>	<b>T</b>		<b>P</b>	
<b>3. Course Code</b>		<b>3</b>	<b>0</b>		<b>2</b>	
<b>4. Type of Course (use tick mark)</b>		<b>Core (√)</b>	<b>PE()</b>		<b>OE()</b>	
<b>5. Pre-requisite (if any)</b>		<b>6. Frequency (use tick marks)</b>	<b>Even (√)</b>	<b>Odd (√)</b>	<b>Either Sem ()</b>	<b>Every Sem ()</b>
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 24</b>		

<b>Course Description:</b>		
The structure and components of computer networks, packet switching, layered architectures, TCP/IP, physical layer, error control, window flow control, local area networks (Ethernet, Token Ring; FDDI), network layer, congestion control, quality of service, multicast		
<b>Learning objectives:</b>		
<ol style="list-style-type: none"> <li>1. Discuss the evolution of computer network concepts.</li> <li>2. Understand the structure of computer networks, factors affecting computer network deployment.</li> <li>3. Describe emerging technology in the net-centric computing area and assess their current capabilities, limitations and potential applications.</li> <li>4. Program and analyse network protocols, architecture, algorithms and other safety critical issues in real-life scenario.</li> </ol>		
<b>Course Outcomes:</b>		
<ol style="list-style-type: none"> <li>1. Examine and analyze various protocols like transport-layer concepts: Transport-Layer services</li> <li>2. -Reliable vs. un-reliable data transfer -TCP protocol -UDP protocol</li> <li>3. Examine and analyze the network-layer concepts like Network-Layer services –Routing -IP protocol -IP addressing</li> <li>4. Examine and analyze the different link-layer and local area network concepts like Link-Layer services –Ethernet -Token Ring -Error detection and correction -ARP protocol</li> <li>5. Analyze and implement application of network system.</li> </ol>		
<b>11. Unit wise detailed content</b>		
<b>Unit-1</b>	<b>Number of lectures = 10</b>	<b>Title of the unit: Introduction Concepts</b>
Goals and Applications of Networks, Network structure and architecture, The OSI reference model, services, Network Topology Design - Delay Analysis, Back Bone Design, Local Access Network Design, Physical Layer Transmission Media, Switching methods, ISDN, Terminal Handling.		
<b>Unit - 2</b>	<b>Number of lectures = 8</b>	<b>Title of the unit: Medium Access sub layer</b>
Medium Access sub layer - Channel Allocations, LAN protocols -ALOHA protocols - Overview of IEEE standards - FDDI. Data Link Layer - Elementary Data Link Protocols, Sliding Window protocols, Error Handling.		
<b>Unit - 3</b>	<b>Number of lectures = 8</b>	<b>Title of the unit: Network Layer</b>

Network Layer - Point - to Point Networks, routing, Congestion control Internetworking -TCP / IP, IP packet, IP address, IPv6.		
<b>Unit - 4</b>	<b>Number of lectures = 8</b>	<b>Title of the unit: Transport Layer</b>
Transport Layer - Design issues, connection management, session Layer-Design issues, remote procedure call. Presentation Layer-Design issues, Data compression techniques, cryptography - TCP - Window Management. Application Layer: File Transfer, Access and Management, Electronic mail, Virtual Terminals, Other application. Example Networks- Internet and Public Networks.		
<b>11. Brief Description of self learning / E-learning component</b> <b>The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University.</b> <b>The link to the E-Learning portal.</b>  <a href="https://elearning.sgtuniversity.ac.in/course-category/">https://elearning.sgtuniversity.ac.in/course-category/</a>		
<b>13. Text Books Recommended</b> Reference Books <ul style="list-style-type: none"> <li>I. —Data Communication and Networking  by B. A. Forouzen, TMH, 4<sup>th</sup> Edition, 2017</li> <li>II. . Computer Networks, A.S. Tanenbaum,Pearson Education, 5<sup>th</sup> Edition, 2013</li> <li>III. Data and Computer Communication, W. Stallings, Pearson Education, 10th Edition, 2013</li> <li>IV. Essential of TCP/ IP  G. Shanmugarathinam, Firewall Media, 2008</li> </ul>		

**Semester I**  
**Problem Solving and Programming with C++(OOPS) Lab**

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	<b>Problem Solving and Programming with C++(OOPS) Lab</b>	<b>L</b>	<b>T</b>	<b>P</b>		
<b>3. Course Code</b>		<b>0</b>	<b>0</b>	<b>2</b>		
<b>4. Type of Course (use tick mark)</b>		Core (✓)	PE()		OE ()	
<b>5. Pre-requisite (if any)</b>	<b>Programming in 'C' &amp; Data Structure</b>	<b>6. Frequency (use tick marks)</b>	<b>Even</b>	<b>Odd (✓)</b>	<b>Either Sem ()</b>	<b>Every Sem ()</b>
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 0</b>		<b>Tutorials = 0</b>	<b>Practical = 48</b>			
<b>8. Course Description:</b> This lab course provides in-depth coverage of objectoriented programming principles and techniques using C++. Topics include classes, overloading, data abstraction, information hiding, encapsulation, inheritance, polymorphism, file processing, templates, exceptions, container classes, and low-level language features.						
<b>9. Learning objectives:</b>						
1. To understand fundamentals of programming such as variables, conditional and iterative execution, methods, etc.						
2. To understand fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries, etc						
3. To have the ability to write a computer program to solve specified problems						
<b>10. Course Outcomes (COs):</b>						
1. Understand the features of C++ supporting object oriented programming						
2. Understand the relative merits of C++ as an object oriented programming language						
3. Understand the features of C++ supporting object oriented programming						
4. Understand advanced features of C++ specifically stream I/O, templates and operator overloading						
5. Understand the relative merits of C++ as an object oriented programming language						
<b>11. List of Experiments</b>						
1. Simple C++ programs to implement various control structures.						
a. if statement						
b. switch case statement and do while loop						
c. for loop						
d. while loop						
2. Programs to understand structure & unions.						
a. structure						
b. union						
3. Programs to understand pointer arithmetic.						
4. Functions & Recursion.						
a. recursion b. function						
5. Inline functions.						
6. Programs to understand different function call mechanism.						
a. call by reference b. call by value						

7. Programs to understand storage specifiers.
8. Constructors & destructors.
9. Use of “this” pointer using class
10. Programs to implement inheritance and function overriding.
  - a. multiple inheritance –access Specifiers
  - b. hierarchical inheritance – function overriding /virtual Function
11. Programs to overload unary & binary operators as member function & non member function.
  - a. unary operator as member function
  - b. binary operator as non member function
12. Programs to understand friend function & friend Class.
  - a. friend Function
  - b. friend class
13. Programs on class templates
14. Using a C++ program check whether a student passed the exam or not based on total mark which shall be above 40%
15. Create a C++ program which takes two distances in an inch-feet system and stores in data members of two structure variables. Then, this program calculates the sum of two distances and displays it.

During the course student must be do project on:

1. Tic Tac Toe Game Project (This project will be without graphics which focus on logic /algorithm used in game. Two players can play this game.)
2. Supermarket Billing Project (Student can build product class with data members like product no, product name, price, qty, tax, discount. Product details is stored in a binary file. A customer can purchase a product and his invoice generated. Administrator can create, modify, view and delete product record.)
3. Fortune Teller Project in C++ (Student can build console application which can determine the horoscope, and predict the future of user based on date of birth, name, and sex entered)
4. Search Engine (Student can build an academic search engine application which is designed to search relevant academic information and records in schools, colleges and universities)

At least one Project is mandatory for each student.

## **12. Brief Description of self-learning / E-learning component**

<http://vlabs.iitb.ac.in/vlabs-dev/labs/oops/index.php>

**Semester I**  
**SOFTWARE TESTING LAB**

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	<b>SOFTWARE TESTING LAB</b>	L	T	P		
<b>3. Course Code</b>		0	0	2		
<b>4. Type of Course (use tick mark)</b>		Core (✓)	PE()	OE ()		
<b>5. Pre-requisite (if any)</b>	<b>Software Engineering</b>	<b>6. Frequency (use tick marks)</b>	Even	Odd (✓)	Either Sem ()	Every Sem ()
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 0</b>		<b>Tutorials = 0</b>	<b>Practical = 48</b>			
<b>8. Course Description</b>						
<b>9. Learning objectives:</b>						
<ol style="list-style-type: none"> <li>1. To apply software testing knowledge and engineering methods.</li> <li>2. Have an ability to design and conduct a software test process for a software testing project.</li> <li>3. To identify the needs of software test automation, and define and develop a test tool to support test automation.</li> <li>4. Have an ability understand and identify various software testing problems, and solve these problems by designing and selecting software test models, criteria, strategies, and methods.</li> <li>5. Have basic understanding and knowledge of contemporary issues in software testing, such as component-based software testing problems</li> <li>6. To use software testing methods and modern software testing tools for their testing projects.</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
<ol style="list-style-type: none"> <li>1. Understand fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods.</li> <li>2. Gain software testing experience by applying software testing knowledge and methods to practice-oriented software testing projects.</li> <li>3. Understand software test automation problems and solution</li> <li>4. Comprehend the concepts related to Software Quality Attributes, Quality Planning, Software Quality Control and Software Quality Assurance.</li> </ol>						
<b>11. List of Experiments</b>						
<ol style="list-style-type: none"> <li>1) Write programs in C language to demonstrate the working of the following constructs:               <ol style="list-style-type: none"> <li>i) do...while</li> <li>ii) while .. do</li> <li>iii) if.. else</li> <li>iv) switch</li> <li>v) for</li> </ol> </li> <li>2) A program written in C language for Matrix Multiplication fails! Introspect the causes for its failure and write down the possible reasons for its failure.</li> <li>3) Take any system (e.g. ATM system) and study its system specifications and report the various bugs.</li> <li>4) Write the test cases for any known application(e.g. Banking application)</li> <li>5) Create a test plan document for any application (e.g. Library Management System)</li> <li>6) Study of any testing tool (e.g. Winrunner)</li> <li>7) Study of any web testing tool (e.g. Selenium)</li> <li>8) Study of any bug tracking tool (e.g. Bugzilla, bugbit)</li> <li>9) Study of any test management tool (e.g. Test Director)</li> <li>10) Study of any open source-testing tool (e.g. Test Link)</li> </ol> <p>At least one Project is mandatory for each student.</p>						
<b>12. Brief Description of self-learning / E-learning component</b>						
<a href="http://vlabs.iitb.ac.in/vlabs-dev/labs/oops/index.php">http://vlabs.iitb.ac.in/vlabs-dev/labs/oops/index.php</a>						



## Semester I

### Networking technologies Lab

<b>13. Name of the Department- Computer Science &amp; Engineering</b>						
<b>14. Course Name</b>	<b>Networking technologies Lab</b>	<b>L</b>	<b>T</b>		<b>P</b>	
<b>15. Course Code</b>		<b>0</b>	0		2	
<b>16. Type of Course (use tick mark)</b>		Core (✓)	PE()		OE ()	
<b>17. Pre-requisite (if any)</b>	<b>Software Engineering</b>	<b>18. Frequency (use tick marks)</b>	Even	Odd (✓)	Either Sem ()	Every Sem ()
<b>19. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 0</b>		<b>Tutorials = 0</b>	Practical = 48			
<b>20. Course Description</b>						
<b>21. Learning objectives:</b>						
<ol style="list-style-type: none"> <li>1. Discuss the evolution of computer network concepts.</li> <li>2. Understand the structure of computer networks, factors affecting computer network deployment.</li> <li>3. Describe emerging technology in the net-centric computing area and assess their current capabilities, limitations and potential applications.</li> <li>4. Program and analyse network protocols, architecture, algorithms and other safety critical issues in real-life scenario.</li> </ol>						
<b>22. Course Outcomes (COs):</b>						
<ol style="list-style-type: none"> <li>1. Examine and analyze various protocols like transport-layer concepts: Transport-Layer services - Reliable vs. un-reliable data transfer -TCP protocol -UDP protocol</li> <li>2. Examine and analyze the network-layer concepts like Network-Layer services –Routing -IP protocol -IP addressing</li> <li>3. Analyze and implement application of network system.</li> </ol>						
<b>23. List of Experiments</b>						
<ol style="list-style-type: none"> <li>1. To learn handling and configuration of networking hardware like RJ-45 connector, CAT-6 cable, crimping tool, etc.</li> <li>2. Configuration of router, hub, switch etc. (using real devices or simulators)</li> <li>3. Running and using services/commands like ping, traceroute, nslookup, arp, telnet, ftp, etc.</li> <li>4. Network packet analysis using tools like Wireshark, tcpdump, etc.</li> <li>5. Network simulation using tools like Cisco Packet Tracer, NetSim, OMNeT++, NS2, NS3, etc.</li> <li>6. Socket programming using UDP and TCP (e.g., simple DNS, data &amp; time client/server, echo client/server, iterative &amp; concurrent servers)</li> <li>7. Programming using raw sockets</li> <li>8. Programming using RPC</li> </ol> <p>At least one Project is mandatory for each student.</p>						
<b>24. Brief Description of self-learning / E-learning component</b>						

**Semester I  
MCA**

Name of the Department- Computer Science & Engineering						
<b>Course Name</b>	<b>Advanced Database Management System Lab</b>	<b>L</b>	<b>T</b>		<b>P</b>	
<b>Course Code</b>		<b>0</b>	<b>0</b>		<b>2</b>	
<b>Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE()</b>		<b>OE ()</b>	
<b>Pre-requisite (if any)</b>		<b>Frequency (use tick marks)</b>	<b>Even</b>	<b>Odd (✓)</b>	<b>Either Sem ()</b>	<b>Every Sem ()</b>
<b>Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 0</b>		<b>Tutorials = 0</b>	<b>Practical = 24</b>			
<b>Course Description</b>						
<ol style="list-style-type: none"> <li>1. To explore the features of a Database Management Systems</li> <li>2. To interface a database with front end tools</li> <li>3. To understand the internals of a database system</li> </ol>						
<b>Course Outcomes (COs):</b>						
<ol style="list-style-type: none"> <li>1. Ability to use databases for building web applications.</li> <li>2. Gaining knowledge about the internals of a database system.</li> <li>3. To gain the knowledge of PHP</li> <li>4. Ability to use Big data analytics.</li> </ol>						
<b>List of Experiments</b>						
<ol style="list-style-type: none"> <li>1. Experiments should be Project Oriented</li> <li>2. Basic SQL</li> <li>3. Intermediate SQL</li> <li>4. Advanced SQL</li> <li>5. ER Modelling</li> <li>6. Database Design and Normalization</li> <li>7. Accessing Databases from Programs using JDBC</li> <li>8. Building Web Applications using PHP &amp; MySQL</li> <li>9. Indexing and Query Processing</li> <li>10. Query Evaluation Plans</li> <li>11. Concurrency and Transactions</li> <li>12. Big Data Analytics using Hadoop</li> <li>13. During the course student must be do project on:</li> <li>14. E- Commerce Management (Student can build an E-commerce platform, where a customer can register and buy a product)</li> <li>15. Inventory Management (Student can build a project which can Increase the inventory turnover for any business)</li> <li>16. Solution for Saving Student Records (Student can build a solution that saves student records for an educational institution)</li> <li>17. Payroll Management Solution (create a database solution for managing payroll)</li> <li>18. At least one Project is mandatory for each student.</li> </ol>						
<b>Brief Description of self-learning / E-learning component</b>						
<p>The students will be encouraged to learn using Virtual Link. Please add VLink  <a href="http://vlabs.iitb.ac.in/bootcamp/labs/dbms/exp8/exp/index.php">http://vlabs.iitb.ac.in/bootcamp/labs/dbms/exp8/exp/index.php</a></p>						

**Semester II**  
**Computer Organization & Architecture**

<b>1. Name of the Department: CSE</b>						
<b>2. Course Name</b>	<b>Computer Organization &amp; Architecture</b>	<b>L</b>	<b>T</b>	<b>P</b>		
<b>3. Course Code</b>		<b>3</b>	<b>0</b>	<b>2</b>		
<b>4. Type of Course (use tick mark)</b>		<b>Core (√)</b>	<b>PE()</b>		<b>OE()</b>	
<b>5. Pre-requisite (if any)</b>		<b>6. Frequency (use tick marks)</b>	<b>Even (√)</b>	<b>Odd ()</b>	<b>Either Sem ()</b>	<b>Every Sem ()</b>
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 0</b>		

<b>Course Description:</b> Introduction to organizational Basic building block diagram of a digital computer system. As the course progresses each major block ranging from Processor to I/O will be discussed in their full architectural detail. The course talks primarily about Computer Organization and Architecture issues, Architecture of a typical Processor, Memory Organization, I/O devices and their interface and System Bus organization etc.						
<b>Learning objectives:</b> <ol style="list-style-type: none"> <li>1. To understand the basic knowledge of Computer system and its component and functioning of each components.</li> <li>2. To understand and analyze computer architecture and organization, computer arithmetic, and CPU design.</li> <li>3. To understand I/O system interconnection</li> <li>4. To understand Memory organization.</li> </ol>						
<b>Course Outcomes:</b> <ol style="list-style-type: none"> <li>1. Able to understand Computer system and its component and functioning of each components.</li> <li>2. To analyze computer architecture and organization, computer arithmetic, and CPU design.</li> <li>3. To understand I/O system and interconnection structures of computer system.</li> <li>4. To understand and analyze I/O techniques and functioning of memory.</li> <li>5. To understand various types of buses in a computer system and illustrate how data transfers is performed.</li> </ol>						
<b>11. Unit wise detailed content</b>						
<b>Unit-1</b>	<b>Number of lectures = 10</b>					
<p><b>Information Representation:</b> Number systems, BCD codes, character codes, error detecting and correcting codes, fixed-point and floating point representation of information. Binary arithmetic operations.</p> <p><b>Binary Logic:</b> Logic gates, Boolean algebra, Boolean functions, truth tables, canonical and standard forms, simplification of Boolean functions.</p> <p><b>Combinational Logic circuits:</b> Design procedure, adders, subtractors, encoders, decoders, multiplexers and demultiplexers.</p>						
<b>Unit - 2</b>	<b>Number of lectures = 8</b>					
<p><b>Sequential Logic circuits:</b> Flip-flops: R-S, J-K FF, Master slave J-K FF, D FF, T FF, computer registers.</p> <p><b>Counters:</b> Synchronous, Asynchronous, updown and programmable counters.</p>						

<b>Unit - 3</b>	<b>Number of lectures = 8</b>	<b>Title of the unit: Network Layer</b>
<p><b>CPU organization:</b> Processor organization, Machine instructions, instruction cycles, memory reference instructions, stack organization, instruction formats and addressing modes, microprogramming concepts, Types of interrupts.</p> <p><b>I/O Organization:</b> I/O interface, interrupt structure, transfer of information between CPU/memory and I/O devices, DMA and IOP.</p>		
<b>Unit - 4</b>	<b>Number of lectures = 8</b>	
<p><b>Basic concepts</b> – Semiconductor RAMs - ROMs – Speed - size and cost – Cache memories - Performance consideration – Virtual memory- Memory Management requirements – Secondary storage, Accessing I/O devices – Interrupts – Direct Memory Access – Buses – Interface circuits – Standard I/O Interfaces (PCI, SCSI, USB).</p>		
<p>11. Brief Description of self learning / E-learning component  The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University.  The link to the E-Learning portal.</p> <p><a href="https://elearning.sgtuniversity.ac.in/course-category/">https://elearning.sgtuniversity.ac.in/course-category/</a></p>		
<p>13. Text Books Recommended  Computer Organization and Architecture – Designing for Performance - William Stallings, Pearson Education, 9<sup>th</sup> Edition, 2012.</p> <ul style="list-style-type: none"> <li>Digital Logic and Computer Design- Mano, M. Morris, Pearson Education First Edition, 2016.</li> </ul> <p>Reference Books  Computer Organization - Carl Hamacher, Zvonko Vranesic and Safwat Zaky, 5th Edition, McGraw-Hill, 2011  Computer Organisation and Design - Patterson, Elsevier Pub., 4<sup>th</sup> Edition, 2011  Computer Organization and Design: The hardware / software interface - David A.Patterson  Computer Architecture and Organization - John P.Hayes, Tata McGraw Hill,3<sup>rd</sup> Edition,2017..</p>		

**Semester II  
Advanced Java**

Name of the Department- Computer Science & Engineering						
<b>2. Course Name</b>	<b>Advanced Java</b>	<b>L</b>	<b>T</b>		<b>P</b>	
<b>3. Course Code (13470204)</b>		<b>3</b>	<b>0</b>		<b>0</b>	
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE()</b>		<b>OE ()</b>	
<b>5. Pre-requisite (if any)</b>	<b>Basic knowledge of C programming language.</b>	<b>6. Frequency (use tick marks)</b>	<b>Even (✓)</b>	<b>Odd ()</b>	<b>Either Sem ()</b>	<b>Every Sem ()</b>
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 0</b>		
<b>8. Course Description</b>						
<p>The revolution in IT (Information Technology) is possible due to evolution of programming languages over the time. With the time, the programming languages become more simple, object oriented, robust and secure to use. Java is one of the programming language that imbibes all the above mentioned features and also, it is used to develop mobile, desktop GUI, web-based, cloud computing applications. This course aims to cover the advance concept of java programming language which includes network programming, database programming, and servlets.</p>						
<b>9. Learning objectives:</b>						
<ol style="list-style-type: none"> <li>1. To understand the fundamentals of object-oriented programming in java, which includes the definition of classes, methods and use of java libraries.</li> <li>2. To understand the application of java programming language in advance applications.</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
<ol style="list-style-type: none"> <li>1. Understanding the structure and model of the java programming language.</li> <li>2. Using java programming language to develop various applications.</li> <li>3. Develop software using java programming language.</li> </ol>						
<b>11. Unit wise detailed content</b>						
<b>Unit-1</b>	<b>Number of lectures = 10</b>					
<p>Introducing classes, objects and methods: defining Class Fundamentals, Object &amp; Object reference, Object Life time &amp; Garbage Collection, Creating and Operating Objects, Constructor &amp; initialization code block, Access Control, Modifiers, methods Nested , Inner Class &amp; Anonymous Classes , Abstract Class &amp; Interfaces Defining Methods, Argument Passing Mechanism , Method Overloading, Recursion, Dealing with Static Members, Finalize() Method, Native Method. Use of “this” reference, Use of Modifiers with Classes &amp; Methods, Design of Accessors and Mutator Methods Cloning Objects, shallow and deep cloning, Generic Class Types.</p> <p>Extending Classes and Inheritance: Use and Benefits of Inheritance in OOP, Types of Inheritance in Java, Inheriting Data members and Methods, Role of Constructors in inheritance, Overriding Super Class Methods, Use of “super”, Polymorphism in inheritance, Type Compatibility and Conversion Implementing interfaces.</p>						
<b>Unit – 2</b>	<b>Number of lectures = 9</b>					
<p>Thread: Understanding Threads, Needs of Multi-Threaded Programming, Thread Life-Cycle, Thread Priorities ,Synchronizing Threads, Inter Communication of Threads ,Critical Factor in Thread – Deadlock.</p>						

GUI Programming: Designing Graphical User Interfaces in Java, Components and Containers, Basics of Components, Using Containers, Layout Managers, AWT Components, Adding a Menu to Window, Extending GUI Features Using Swing Components, Java Utilities (java.util Package) The Collection Framework : Collections of Objects , Collection Types, Sets , Sequence, Map, Understanding Hashing, Use of ArrayList & Vector.		
<b>Unit – 3</b>	<b>Number of lectures</b> <b>= 8</b>	
Event Handling: Event-Driven Programming in Java, Event- Handling Process, Event-Handling Mechanism, The Delegation Model of Event Handling, Event Classes, Event Sources, Event Listeners, Adapter Classes as Helper Classes in Event Handling. Network Programming: Socket based communication & Remote method invocation (RMI).		
<b>Unit – 4</b>	<b>Number of lectures</b> <b>= 9</b>	
Database Programming using JDBC: Introduction to JDBC, JDBC Drivers & Architecture, CRUD operation Using JDBC, Connecting to non-conventional Databases. Java Server Technologies (Servlet): Web Application Basics, Architecture and challenges of Web Application, Introduction to servlet, Servlet life cycle, Developing and Deploying Servlets, Exploring Deployment, Descriptor (web.xml), Handling Request and Response.		
12. Brief Description of self-learning / E-learning component The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University. The link to the E-Learning portal. <a href="https://elearning.sgtuniversity.ac.in/course-category/">https://elearning.sgtuniversity.ac.in/course-category/</a>		
<b>13. Books Recommended</b>		
Text Books Java, Herbert Schildt. "The Complete Reference." Complete Reference Series) 10th Edition  New York: McGraw-Hill Education (2017).		
<b>Reference Books</b>		
SAMANTA, DEBASIS. Object-oriented Programming with C++ and Java. PHI Learning Pvt. Ltd., 2006.. <a href="https://cse.iitkgp.ac.in/~dsamanta/java/index.htm">https://cse.iitkgp.ac.in/~dsamanta/java/index.htm</a> , <a href="https://nptel.ac.in/courses/106/105/106105191/">https://nptel.ac.in/courses/106/105/106105191/</a> E. Balaguruswamy, "Programming with Java: A Primer", McGraw-Hill; Sixth edition, 2019.		

**Semester II**  
**Medical Image Processing**

Name of the Department- Computer Science & Engineering						
<b>2. Course Name</b>	<b>Medical Image Processing</b>	<b>L</b>	<b>T</b>		<b>P</b>	
<b>3. Course Code</b>		<b>3</b>	<b>0</b>		<b>0</b>	
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE()</b>		<b>OE ()</b>	
<b>5. Pre-requisite (if any)</b>	<b>Basic knowledge of C programming language.</b>	<b>6. Frequency (use tick marks)</b>	<b>Even (✓)</b>	<b>Odd ()</b>	<b>Either Sem ()</b>	<b>Every Sem ()</b>
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 0</b>		
<b>8. Course Description</b>						
The purpose of this course is to learn the fundamentals and various techniques of medical image processing and to develop the algorithms for image analysis and diagnosis in medical imaging.						
<b>9. Learning objectives:</b>						
<ol style="list-style-type: none"> <li>1. study of methods for enhancing, analyzing, interpreting</li> <li>2. visualizing information from two- and three-dimensional data obtained from a variety of medical imaging modalities.</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
<ol style="list-style-type: none"> <li>1. Identify major processes involved in formation of medical images</li> <li>2. Recognize the imaging modality from their visualization</li> <li>3. Classify the various medical image processing algorithms</li> <li>4. Describe fundamental methods of image enhancement</li> </ol>						
<b>11. Unit wise detailed content</b>						
<b>Unit-1</b>	<b>Number of lectures = 10</b>					
Fundamentals of Digital image Image formation, visual perception, CCD & CMOS Image sensor, Image sampling: Two dimensional Sampling theory, Nonrectangular grid and Hexagonal sampling, Optimal sampling, Image quantization, Non uniform Quantization, Image formats. Types of pixel Operations, Types of neighborhoods, adjacency, connectivity, boundaries, regions, 2Dconvolution, Color models.						
<b>Unit – 2</b>	<b>Number of lectures = 9</b>					
Image Enhancement in Spatial and Frequency domain Basic gray level transformations, histogram processing, Smoothing operations, Edge Detection-derivative based operation, filtering in frequency domain, 2D-DFT, Smoothing frequency domain filters, Sharpening frequency domain filters, Homomorphic filtering.						
<b>Unit – 3</b>	<b>Number of lectures = 8</b>					
Morphological Image Processing Dilation and Erosion, Opening and Closing, Hit-or-Miss transformation, Boundary Extraction, Region filling, Extraction of Connected Components, Convex Hull, Thinning, Thickening,						

Skeletons, Pruning. Image Segmentation Detection of discontinuities, Point-line- edge detection, Linear and Circular Hough Transform, Basic Global and Adaptive Thresholding, Region Based segmentation, K-Means Clustering	
<b>Unit – 4</b>	<b>Number of lectures = 9</b>
Image Compression: Fundamentals of Image compression models, Lossless compression: variable length coding, LZW coding, Arithmetic coding, Lossy compression: Wavelet and DCT coding, Predictive coding. Representation and Description Image features, Feature extraction, Chain code, Moments	
<b>12. Brief Description of self-learning / E-learning component</b> The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University. The link to the E-Learning portal.	
<b>13. Books Recommended</b>	
<b>Text Books</b> 1. Digital Image Processing, Gonzalez and Woods- Pearson Education 2. Digital Image Processing, S. Sridhar – Oxford University Press.	
<b>Reference Books</b> 1. Fundamentals of Digital Image Processing, A.K. Jain .P.H.I. 2. Digital Image Processing, William Pratt- John Wiley. 3. Feature Extraction and Image Processing, Mark S. Nixon and Alberto S. Aguado. 4. Digital Image Processing and Analysis, Chanda Majumder- Printice Hall India. 5. 7. Medical image processing, Geoff Dougherty edito	



**Semester II**  
**Advanced Operating System**

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	<b>Advanced Operating System</b>	<b>L</b>	<b>T</b>		<b>P</b>	
<b>3. Course Code</b>		<b>3</b>	<b>0</b>		<b>0</b>	
<b>4. Type of Course (use tick mark)</b>		Core (✓)	<b>PE()</b>		<b>OE ()</b>	
<b>5. Pre-requisite (if any)</b>	Operating System	<b>6. Frequency (use tick marks)</b>	Even (✓)	Odd ()	Either Sem ()	Every Sem ()
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 0</b>		
<b>8. Course Description</b>						
Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.						
<b>9. Learning objectives:</b>						
1. To impart the knowledge on the need and requirement of an interface between Man and Machine.						
2. To teach the features of operating systems and the fundamental theory associated with process, memory and file management components of operating systems.						
3. To understand the memory Architecture						
<b>10. Course Outcomes (COs):</b>						
The students will be able to						
<ul style="list-style-type: none"> <li>● Describe the general architecture of computers</li> <li>● Describe, contrast and compare differing structures for operating systems</li> <li>● Understand and analyse theory and implementation of: processes, resource control (concurrency etc.), physical and virtual memory, scheduling, I/O and files</li> </ul>						
<b>11. Unit wise detailed content</b>						
<b>Unit-1</b>	<b>Number of lectures = 09</b>					
<p><b>Introduction:</b> Definition, Operating System Structure, Operating System Operations, Distributed system, Methodologies for implementation of O/S service, <b>Processes:</b> Process model, Process states, process hierarchies, implementation of Processes, data structures used such as Process table, PCB creation of processes, context switching, exit of Processes.</p> <p><b>Inter-process communication:</b> Race conditions, critical sections, problems of mutual exclusion, Peterson's solution, producer-customer problem, Reader Writer's Problem, Dining Philosophers Problem, semaphores, monitors, message passing.</p>						
<b>Unit – 2</b>	<b>Number of lectures = 09</b>					
<b>Process scheduling:</b> objective, preemptive vs. non-preemptive scheduling,						

comparative assessment of different algorithms such as round robin, priority bases scheduling. FCFS. SJF, multiple queues with feedback

**Deadlocks:** Conditions, modeling, detection and recovery, deadlock avoidance, deadlock prevention.

**Memory Management:** Swapping, Contiguous Memory Allocation, Paging, Segmentation, Virtual Memory, Demand Paging

**Unit – 3**

**Number of lectures = 08**

File Management- File-System Interface- File Concept- Access Methods – Directory and Disk Structure – File-System Mounting – File Sharing- Protection- File-System Implementation- File System Structure- File-System Implementation- Directory Implementation- Allocation Methods Free-Space Management – Efficiency and Performance

**Unit – 4**

**Number of lectures = 10**

Mass Storage Structure- Disk Scheduling- Disk Management RAID Structure – Stable Storage Implementation- Protection and Security- Protection- Goals of Protection- Principles of Protection Domain of Protection- Access Matrix Implementation of Access Matrix- Access Control- Revocation of Access Rights Security The Security Problem – Program Threats- System and Network Threats.

### **12. Brief Description of self-learning / E-learning component**

The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University.

The link to the E-Learning portal.

<https://elearning.sgtuniversity.ac.in/course-category/>

### **13. Books Recommended**

#### **Text Books**

- William Stallings, Operating Systems: Internals and Design Principles, 6 th Ed., Pearson Education

### **14. Reference Books**

- Nutt G.J., Operating Systems, 3 rd Ed., Pearson Education.
- Silberschatz, Galvin, & Gagne, Operating System Concepts, 8 th Ed., Wiley
- Tanenbaum A.S., Modern Operating Systems, 3 rd Ed., Prentice Hall

**Semester II**  
**Data Structures and Algorithm Design**

<b>Name of the Department- Computer Science &amp; Engineering</b>						
<b>Course Name</b>	<b>Data Structures and Algorithm Design</b>	<b>L</b>	<b>T</b>		<b>P</b>	
<b>Course Code</b>		<b>3</b>	<b>0</b>		<b>0</b>	
<b>Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE()</b>		<b>OE ()</b>	
<b>Pre-requisite (if any)</b>		<b>Frequency (use tick marks)</b>	<b>Even (✓)</b>	<b>Odd ()</b>	<b>Either Sem ()</b>	<b>Every Sem ()</b>
<b>Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 0</b>		
<b>Course Description</b>						
<p>Topics include elementary data structures, (including arrays, stacks, queues, and lists), advanced data structures (including trees and graphs), the algorithms used to manipulate these structures, and their application to solving practical engineering problems.</p> <p>Students study techniques for designing algorithms and for analyzing the time and space efficiency of algorithms. The algorithm design techniques include divide-and-conquer, greedy algorithms, dynamic programming.</p>						
<b>Learning objectives:</b>						
<ol style="list-style-type: none"> <li>1. Learn new data &amp; file structure concepts and algorithms.</li> <li>2. Learn when and where these concepts would be used in real-world applications and programming contexts.</li> <li>3. Apply many of these concepts/algorithms by using them in programming projects.</li> <li>4. Date Structures are an integral part of algorithm design and Discrete Structures covers topics like graph theory.</li> </ol>						
<b>Course Outcomes (COs):</b>						
<ol style="list-style-type: none"> <li>1. Understand of the basic data structures.</li> <li>2. The appropriate use of a particular data structure and algorithm to solve a problem.</li> <li>3. Demonstrate a familiarity with major algorithms and data structures.</li> <li>4. Apply important algorithmic design paradigms and methods of analysis.</li> <li>5. Synthesize efficient algorithms in common engineering design situations.</li> </ol>						
Unit wise detailed content						
<b>Unit-1</b>	<b>Number of lectures = 09</b>					
<p>Introduction to Data Structure: Data types, Abstract Data types, Arrays, Arrays as abstract data type, Arrays row major and column major, Sequences, Big Oh notations. Stack: Definition and Example, Representing Stack using static implementation, Applications, Infix, Prefix and postfix, Converting infix to postfix Expression, Evaluation Matching parentheses, Recursion and Simulating Recursion. Queues: Definition and examples, Representing Queues using static implementation, Circular queues, Priority queues, Double-ended queues.</p>						
<b>Unit – 2</b>	<b>Number of lectures = 09</b>					
<p>Linked Lists: List Types (singly, doubly, singly circular, doubly circular), Operations on all types of Lists – create, insert, delete Generalized Lists Applications, Dynamic implementation of stack and queues, Polynomial Addition, Dynamic Memory Allocation – First- Fit, Best – Fit, Worst-fit Trees: Concept Rooted Tree Binary Tree–Linked and static Representation, Tree Traversals (Pre-order, In-order, Post-order using recursion), Binary Search Tree (create, delete, search, insert, display), AVL Trees.</p>						

<b>Unit – 3</b>	<b>Number of lectures = 08</b>	
<p>Introduction: Algorithm, Analyzing algorithm, Designing algorithm, Concept of algorithmic efficiency, Run time analysis of algorithms, Asymptotic Notations.</p> <p>Divide and Conquer: Structure of divide and conquer algorithms; examples; Greedy Method: Overview of the Greedy Paradigm, Examples of Exact Optimization solution (minimum cost spanning tree) Dynamic Programming: Overview, Difference between Dynamic Programming and Divide and Conquer, Applications: Shortest path in graph, Traveling salesman Problem.</p>		
<b>Unit – 4</b>	<b>Number of lectures = 10</b>	
<p>Back Tracking: Overview, 8-queen problem, Graph Coloring Problem and Knapsack problem</p> <p>Graphs: Representation using C Adjacency matrix and adjacency lists BFS and DFS bystatic and dynamic implementation, Finding shortest path (Dijkstra’s Algorithm) Searching: Sequential, Binary, Hashing, Hash tables, Hash functions, Overflow handling techniques.</p> <p>Sorting: Bubble sort, Insertion sort, Quick sort (recursive), Merge sort, Heap sort and Bucket sort.</p> <p>Complexity measures, Polynomial vs. non-polynomial time complexity; NP-hard and NP-complete classes, examples.</p>		
<p>Brief Description of self-learning / E-learning component</p> <p>The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University.</p> <p>The link to the E-Learning portal.</p> <p><a href="https://elearning.sgtuniversity.ac.in/course-category/">https://elearning.sgtuniversity.ac.in/course-category/</a></p>		
<b>Books Recommended</b>		
<p>Text Books</p> <p>Data Structures with C (Schaum's Outline Series), Seymour Lipschutz, McGraw Hill Education; 1st edition, 2017.</p> <p>Introduction of Computer Algorithm, T. H Cormen, Leiserson, Rivest and Stein, PHI, New Delhi.</p> <p>Fundamentals of Computer Algorithms. 2<sup>nd</sup> Edition, E. Horowitz, S. Sahni, and S.Rajsekran, University Press, Hyderabad.</p>		
<b>Reference Books</b>		
<p>Computer Algorithms, Sara Basse, A.V. Gilder, Addison Wesley, New Delhi.</p>		
<p>Fundamentals of Data Structure, E. Horowitz, S. Sahni, and S.Rajsekran University Press, Hyderabad</p>		
<p>Data Structures Using C, Balagurusamy, McGraw Hill Education; First edition, 2016</p>		

**Semester II**  
**Advanced Operating System Lab**

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	<b>Advanced Operating System Lab</b>	<b>L</b>	<b>T</b>	<b>P</b>		
<b>3. Course Code</b>		<b>0</b>	<b>0</b>	<b>2</b>		
<b>4. Type of Course (use tick mark)</b>		Core (✓)	PE()	OE ()		
<b>5. Pre-requisite (if any)</b>		<b>6. Frequency (use tick marks)</b>	Even (✓)	Odd ()	Either Sem ()	Every Sem ()
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 0</b>		<b>Tutorials = 0</b>		<b>Practical = 24</b>		
<b>8. Course Description</b>						
This course will introduce the core concepts of operating systems, such as processes and threads, scheduling, synchronization, memory management, file systems, input and output device management and security						
<b>9. Learning objectives:</b>						
<ol style="list-style-type: none"> <li>To learn the mechanisms of OS to handle processes and threads and their communication</li> <li>To learn the mechanisms involved in memory management in contemporary OS</li> <li>To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
1. The ability to research, understand and implement computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient analysis and design of computer-based systems of varying complexity						
2. The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success						
3. The ability to employ modern computer languages, environments, and platforms in creating innovative career paths, to be an entrepreneur, and a zest for higher studies						
<b>11. List of Experiments</b>						
<ol style="list-style-type: none"> <li>Write a C program to simulate the FCFS CPU scheduling algorithms to find turnaround time and waiting time.</li> <li>Write a C program to simulate the SJF CPU scheduling algorithms to find turnaround time and waiting time.</li> <li>Write a C program to simulate the Round Robin (preemptive) CPU scheduling algorithms to find turnaround time and waiting time.</li> <li>Write a C program to simulate the Priority CPU scheduling algorithms to find turnaround time and waiting time.</li> <li>Write a C program to simulate the MVT and MFT memory management techniques.</li> <li>Write a C program to simulate the paging technique of memory management.</li> <li>Write a C program to simulate Bankers algorithm for the purpose of deadlock avoidance</li> <li>Write a C program to simulate FIFO page replacement algorithms.</li> <li>Write a C program to simulate LRU page replacement algorithms.</li> <li>Write a C program to simulate LFU page replacement algorithms .</li> <li>Write a C program to simulate producer-consumer problem using semaphores</li> <li>Write a C program to simulate the concept of Dining-Philosophers problem.</li> </ol>						

During the course student must be do project on:

1. The Unix Shell (Student can build project How processes are handled (i.e., starting and waiting for their termination))
2. Web Servers and Synchronization (Student can modify an existing code to learn how to create and synchronize cooperating threads in Unix and gain exposure to how a basic web server is structured)
3. A "Better" Malloc
4. A "File System" (Build a file system using the FUSE library, Keep adding more and more system calls and features as time rolls)

At least one Project is mandatory for each student.

## **12. Brief Description of self-learning / E-learning component**

The students will be encouraged to learn using Virtual Link. Please add VLink

**Semester II**  
**Data Structures and Algorithm Lab**

Name of the Department- Computer Science & Engineering						
Course Name	Data Structures and Algorithm Lab	L	T		P	
Course Code		0	0		2	
Type of Course (use tick mark)		Core (✓)	PE()		OE ()	
Pre-requisite (if any)	Data Structure	Frequency (use tick marks)	Even (✓)	Odd ()	Either Sem ()	Every Sem ()
<b>Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
Lectures = 0		Tutorials = 0	Practical = 24			
<p><b>Course Description</b> The course is designed to develop skills to design and analyze simple linear and non linear data structures. It strengthen the ability to the students to identify and apply the suitable data structure for the given real world problem. It enables them to gain knowledge in practical applications of data structures.</p>						
<p><b>Learning objectives:</b></p> <ol style="list-style-type: none"> <li>To impart the basic concepts of data structures and algorithms</li> <li>To understand concepts about searching and sorting techniques</li> <li>To understand basic concepts about stacks, queues, lists, trees and graphs</li> <li>To understanding about writing algorithms and step by step approach in solving problems with the help of fundamental data structures</li> </ol>						
<b>Course Outcomes (COs):</b>						
1. At the end of this lab session, the student will						
2. Be able to design and analyze the time and space efficiency of the data structure .						
3. Be capable to identity the appropriate data structure for given problem .						
4. Have practical knowledge on the applications of data structures						
<b>List of Experiments</b>						
<ol style="list-style-type: none"> <li>BASIC TECHNIQUES: Swapping of the contents of two variables- Finding the sum of digits of a given number- Reversing a given number.</li> <li>DECISION MAKING: Finding the largest and the smallest of a given array- solving a quadratic equation- selecting an operation based on a menu.</li> <li>LOOPING TECHNIQUES &amp; ARRAYS: Finding the sum to n terms of a sine series- Matrix Multiplication- Transpose-Polynomial addition- Polynomial Multiplication- Sorting algorithms Searching algorithms.</li> <li>CHARACTERS AND STRING HANDLING: Finding the length of string-reversal of string concatenation of two strings-checking whether it is a palindrome or not- converting upper case alphabets to lowercase and vice versa in a string.</li> <li>Implementation of ADT Linked list.</li> <li>Implementation of Stack using arrays.</li> <li>Implementation of Queue using arrays</li> <li>Conversion of prefix expression into post-fix form using stacks.</li> <li>Implementation of Binary tree and its Traversals a)Preorder b)Inorder c)Postorder.</li> <li>Write a C Programming to implement the following Sorting techniques a)Bubblesort b)Quick sort</li> <li>Design, develop and execute a program in C to create a max heap of integers by accepting one element at a time and by inserting it immediately in to heap. Use the array representation of heap. Display the array at the end of insertion phase.</li> </ol>						

- 12.** Design, develop and execute a program in C to implement doubly linked list where each node consist of integers. The program should support following functions.
- 13.** Create a doubly linked list
- 14.** Insert a new node
- 15.** Delete a node if it is found, otherwise display appropriate message
- 16.** Display the nodes of doubly linked list
- 17.** Design, develop and execute a program in C to read a sparse matrix of integer values and make a transpose of it. Use the triple to represent an element in sparse matrix.
- 18.** Design, develop and execute a program in C to implement singly linked list where each node consist of integers. The program should support following functions.
- 19.** Create a singly linked list
- 20.** Insert a new node
- 21.** Delete a node if it is found, otherwise display appropriate message
- 22.** Display the nodes of singly linked list
  
- 23.** During the course student must be do project on:
- 24.** Mini Project in C Personal Diary Management System
- 25.** Typing Tutor Project Using C
- 26.** Mini Calendar Using C
- 27.** Student Database Management and Enquiry System
- 28.** At least one Project is mandatory for each student.



**Semester II**  
**Advance Java lab**

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	<b>Advance Java lab</b>	<b>L</b>	<b>T</b>	<b>P</b>		
<b>3. Course Code</b>		<b>0</b>	<b>0</b>	<b>4</b>		
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE()</b>		<b>OE ()</b>	
<b>5. Pre-requisite (if any)</b>	<b>Basic knowledge of C programming language.</b>	<b>6. Frequency (use tick marks)</b>	<b>Even ()</b>	<b>Odd (✓)</b>	<b>Either Sem ()</b>	<b>Every Sem ()</b>
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 0</b>		<b>Tutorials = 0</b>	<b>Practical = 48</b>			
<b>8. Course Description</b>						
<p>The revolution in IT (Information Technology) is possible due to evolution of programming languages over the time. With the time, the programming languages become simpler, object oriented, robust and secure to use. Java is one of the programming language that imbibes all the above mentioned features and also, it is used to develop mobile, desktop GUI, web-based, cloud computing applications. This course aims to cover the advance concept of java programming language which includes network programming, database programming, and servlets.</p>						
<b>Learning objectives:</b>						
<ol style="list-style-type: none"> <li>1. To understand the fundamentals of object-oriented programming in java, which includes the definition of classes, methods and use of java libraries.</li> <li>2. To understand the application of java programming language in advance applications.</li> </ol>						
<b>10.Course Outcomes (COs):</b>						
<ol style="list-style-type: none"> <li>1. Understanding the structure and model of the java programming language.</li> <li>2. Using java programming language to develop various applications.</li> <li>3. Develop software using java programming language.</li> </ol>						
<b>11. List of Experiments</b>						
1. WAP that describes a class person. It should have instance variables to record name, age and salary. Create a person object. Set and display its instance variables.						
2. Write a program to show the concept of Constructors.						
3. Write an application that shows thread synchronization.						
4. Write an application that displays deadlock between threads.						
5. Write an application that shows thread priorities.						
6. WAP to add label and button in a frame.						
7. WAP to add panel to GUI.						
8. WAP to create a swing button.						
9. WAP to create JFrame, JButton and method call inside the java constructor.						
10. WAP to execute select query using JDBC.						
11. WAP to update client information in the database.						
12. WAP of database connectivity using JDBC-ODBC drivers.						

13. WAP to implement simple servlet that generates the plain text.

14. WAP to display cookie id.

15. WAP to implement socket programming.

16. Write RMI based client-server programs.

**12. Brief Description of self-learning / E-learning component**

<https://www.geeksforgeeks.org/introduction-java-servlets/>

<https://www.geeksforgeeks.org/socket-programming-in-java/>

## Semester III

### Programming in Python

<b>Name of the Department- Computer Science &amp; Engineering</b>						
<b>Course Name</b>	<b>Programming in Python</b>	<b>L</b>	<b>T</b>		<b>P</b>	
<b>Course Code</b>		<b>2</b>	<b>0</b>		<b>0</b>	
<b>Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE()</b>		<b>OE ()</b>	
<b>Pre-requisite (if any)</b>	<b>Basics of Programming</b>	<b>Frequency (use tick marks)</b>	<b>Even ()</b>	<b>Odd (✓)</b>	<b>Either Sem ()</b>	<b>Every Sem ()</b>
<b>Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 0</b>		
<b>Course Description</b>						
An introduction to the Python programming language. Covers details of how to start and stop the interpreter and write programs. Introduces Python's basic datatypes, files, functions, and error handling.						
<b>Learning objectives:</b>						
<ol style="list-style-type: none"> <li>1. To Learn concepts of various Python script at the shell prompt, Python types, expressions to solve relative problems</li> <li>2. To learn different libraries of python</li> <li>3. To learn xml</li> </ol>						
<b>Course Outcomes (COs):</b>						
<ol style="list-style-type: none"> <li>1. To utilize high-level data types such as lists and dictionaries</li> <li>2. To import and utilize a module read from and write to a text file</li> <li>3. Understand the difference between mutable and immutable types</li> <li>4. To demonstration of IDE"s: IDLE, IPython, IPython Notebook, hosted environments</li> </ol>						
<b>Unit wise detailed content</b>						
<b>Unit-1</b>	<b>Number of lectures = 09</b>					
What is Python?, Advantages and disadvantages' Downloading and installing, Which version of Python Running Python Scripts, Using the interpreter interactively.						
<b>Unit – 2</b>	<b>Number of lectures = 09</b>					
Using variables, String types: normal, raw and Unicode, String operators and expressions, Math operators and expressions, Writing to the screen, Reading from the keyboard, Indenting is significant The if and elif statements, While Loops, Using List, Dictionaries, Using the for statement, Opening, reading and writing a text file, Using Pandas, the python data analysis library and data frames, Grouping, aggregating and applying, merging and joining., Dealing with syntax errors, Exceptions, Handling exceptions with try/excep						
<b>Unit – 3</b>	<b>Number of lectures = 09</b>					
RE Pattern Matching, Parsing Data, Introduction to Regression, Types of Regression, Use Cases, Exploratory data analysis, Correlation Matrix, Visualization using Metplotlib, Implementing linear regression						
<b>Unit – 4</b>	<b>Number of lectures = 9</b>					
Machine Learning – Algorithm, Algorithms – Random forest, Super vector Machine, Random						

Forest, Build your own model in python, Comparison between random forest and decision tree
Brief Description of self-learning / E-learning component This learning method gives students to find out their learning capability. Students involve some sort of choice in this learning. As self directed learning learners can determine which modules or scenarios to review again and again. <a href="https://elearning.sgtuniversity.ac.in/course-category/">https://elearning.sgtuniversity.ac.in/course-category/</a>
<b>Books Recommended</b>
<b>Text Books</b> Learning to Program Using Python by Cody Jackson Python for complete beginners by Dr. Martin Jones
<b>Reference Books</b>
Fundamentals of Python: First Programs by Ken Lambert
Learning Python, 5th Edition by Mark Lutz, O'Reilly Media.
Easy GUI Programming in Python by Ken Lambert
The Practice of Computing Using Python by Bill Punch and Rich Enbody

**Semester III**  
**Research Methodology**

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	<b>Research Methodology</b>	<b>L</b>	<b>T</b>	<b>P</b>		
<b>3. Course Code</b>		<b>3</b>	<b>0</b>	<b>0</b>		
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE()</b>	<b>OE ()</b>		
<b>5. Pre-requisite (if any)</b>	<b>+ 2 Mathematics</b>	<b>6. Frequency (use tick marks)</b>	<b>Even ()</b>	<b>Odd (✓)</b>	<b>Either Sem()</b>	<b>Every Sem ()</b>
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 0</b>		
<b>8. Course Description</b>						
Course prelude the foundational methods and techniques of academic research in social sciences and engineering. Engineers would examine and be practically exposed to the main components of a research framework i.e., problem definition, research design, data collection & compilations, report writing, presentation and conclude. Course intended for students requiring hands on knowledge of engineering & sciences applications.						
<b>Learning Objectives:</b>						
<ol style="list-style-type: none"> <li>1. The primary objective of this course is to develop a research orientation among the engineers.</li> <li>2. To provide a foundation for post-secondary education.</li> <li>3. To facilitate the development and application of problem-solving skills in students.</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
<b>The students will be able to:-</b>						
<ol style="list-style-type: none"> <li>1. To develop understanding of the basic framework of research process.</li> <li>2. To develop an understanding of various research designs and techniques.</li> <li>3. To identify various sources of information for literature review and data collection.</li> <li>4. To develop an understanding of the ethical dimensions of conducting applied research.</li> <li>5. To develop an understanding of quality research &amp; scholarly writing.</li> </ol>						
<b>11. Unit wise detailed content</b>						
<b>Unit-1</b>	<b>Number of lectures = 9</b>	<b>Sources and Presentation of Data</b>				
<b>Sources and Presentation of Data</b>						
Statistical Data, Methods of Presentation, Presentation or Illustration of Quantitative Data and Qualitative Data.						
<b>Measures of Location – Averages and Percentiles</b>						
Measure of central tendency – Averages, Measure of Location – Percentiles.						
<b>Unit – 2</b>	<b>Number of lectures = 9</b>	<b>Variability and its Measures</b>				
<b>Variability and its Measures</b>						
Types of Variability, Measures of Variability. Normal Distribution and Normal Curve, Demonstration of a Normal Distribution, Normal curve, Relative or Standard Normal Deviate or Variate (Z).						

<b>Unit – 3</b>	<b>Number of lectures = 9</b>	<b>Correlation and Regression</b>
<b>Correlation and Regression</b> Measures of Relationship between continuous Variables, Types of Correlation, Correlation coefficient from Grouped and Ungrouped series. Regression and Calculation of Regression Coefficient.		
<b>Unit – 4</b>	<b>Number of lectures = 9</b>	<b>Research Methodology</b>
<b>Research Methodology:</b> Meaning of Research, Objective of research, Motivation in research, Types of research, research approaches, research process, & Criteria of good research. Defining the research problems: Selecting the problems, technique involved in defining the problem and conclusion. Research Design: Meaning & Need of research design, different research designs.		
<b>12. Brief Description of self-learning / E-learning component</b> The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University. The link to the E-Learning portal. <a href="https://elearning.sgtuniversity.ac.in/course-category/">https://elearning.sgtuniversity.ac.in/course-category/</a>		
<b>13. Books Recommended</b>		
<b>Text Books</b> C.R. Kothari, Research Methodology, New Age Publications		
<b>14. Reference Books</b>		
SC Gupta & V K Kapoor, Fundamentals of Mathematical Statistics, Sultan Chand & Sons Publications H.C.Saxena, Elementary Statistics, S.Chand Publications. Computers Today, D. H. Sanders, Fourth Edition, McGraw Hill, 1988		

**Semester III**  
**Mobile App Development**

Name of the Department- Computer Science & Engineering						
Course Name	Mobile App Development	L	T		P	
Course Code		3	0		0	
Type of Course (use tick mark)		Core (✓)	PE()		OE ()	
Pre-requisite (if any)	OOPS	Frequency (use tick marks)	Even ()	Odd (✓)	Either Sem ()	Every Sem ()
<b>Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 0</b>		
<b>Course Description</b>						
<p>This course provides a basic understanding of Android development, including the use of content providers, creating audio and video services. This course focuses on helping people become an Android application developer and releasing high-quality apps to the marketplace. Learn about the various stages of development on the Android platform and study topics related to UI, application services, permissions and security, graphics and video resources, data persistence, monitoring tools, mobile app marketing, application hosting and more. Develop core Java development skills while you explore key concepts for building rich applications using advanced features. Learn from instructors and guest speakers working in the industry.</p>						
<b>Learning objectives:</b>						
<ol style="list-style-type: none"> <li>1. learn the set up and installation of Android.</li> <li>2. learn Android App development</li> <li>3. learn user interfaces and Controls.</li> </ol>						
<b>Course Outcomes (COs):</b>						
1. Understand the basics of Android devices and Platform.						
2. Acquire knowledge on basic building blocks of Android programming required for App development						
3. Understand persistence Data storage mechanism in Android						
4. Understand advanced application concepts like networking, Animations and Google Maps services etc						
5. Develop and publish Android applications in to Android Market						
<b>Unit wise detailed content</b>						
<b>Unit- 1</b>	<b>Number of lectures = 09</b>					
<p>Introduction: Introduction to mobile application development, trends, introduction to various platforms, introduction to smart phones</p> <p>Android platform: Android platform features and architecture, versions, comparison added features in each version. ART (Android Runtime), ADB (Android Debug Bridge).</p> <p>Development environment/IDE: Android studio and its working environment, gradle build system, emulator setup</p> <p>Application anatomy: Application framework basics: resources layout, values, asset XML representation and generated R.Javafile</p>						
<b>Unit – 2</b>	<b>Number of lectures = 09</b>					
<p>GUI for Android: Introduction to activities, activities life-cycle, Android v7 support library form API21 for lower version support</p> <p>Intent: intent object, intent filters, adding categories, linking activities, user interface design</p>						

components  
Views and View Groups: Basic views, picker views, adapter views, Menu, App Bar etc, basics of screen design; different layouts. App widgets.  
Lollipop Material design: new themes, new widgets, Card layouts. Recycler View  
Fragments: Introduction to activities, activities life-cycle.

<b>Unit – 3</b>	<b>Number of lectures = 09</b>	
---------------------	------------------------------------	--

Different Data persistence schemes: Shared preferences, File Handling, managing data using SQLite database. Content providers: user content provider, Android in build content providers. Services: introduction to services – local service, remote service and binding the service, the communication between service and activity, Intent Service. Multithreading: Handlers, AsyncTask  
Android network programming: HttpURLConnection, Connecting to REST-based and SOAP based Web services. Broad cast receivers:LocalBroadcastManager, Dynamic broadcast receiver, System Broadcast. PendingIntent, Notifications. Telephony Manager: Sending SMS and making calls.

<b>Unit – 4</b>	<b>Number of lectures = 9</b>	
---------------------	-----------------------------------	--

Location based services: Google maps V2 services using Google API,  
Animations and Graphics: Property Animation, View Animations, Drawable Animations  
Media and Camera API: Working with video and audio inputs, camera API  
Sensor programming: Motion sensors, Position sensors, Environmental sensors.  
Publishing Android Apps: Guide lines, policies and process of uploading Apps to Google play

Brief Description of self-learning / E-learning component  
The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University.  
The link to the E-Learning portal.  
<https://elearning.sgtuniversity.ac.in/course-category/>

**Books Recommended**

**Text Books**  
Dawn Griffiths, David Griffiths,“Head First: Android Development” ,OReilly2015,ISBN: 9781449362188  
David Tainar - Mobile Computing: Concepts Methodologies, Tools & Applications.

**Reference Books**

Barbara L Ciaramtaro - Mobile technology consumption  
Paul Deital,HarveyDeital, Alexander Wald, “Android 6 for Programmers ,App Driven approach”,2015, Prentice Hall ,ISBN: 9780134289366  
<http://developer.android.com/training/index.html> as on Date 21.4.2016



**Semester III**  
**Programming in Python Lab**

Name of the Department- Computer Science & Engineering						
Course Name	Programming in Python Lab	L	T		P	
Course Code		0	0		4	
Type of Course (use tick mark)		Core (✓)	PE()		OE ()	
Pre-requisite (if any)		Frequency (use tick marks)	Even ()	Odd (✓)	Either Sem ()	Every Sem ()
<b>Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
Lectures = 0		Tutorials = 0	Practical = 48			
<b>Course Description</b>						
<b>Learning objectives:</b>						
<ol style="list-style-type: none"> <li>1. To learn concepts of various Python script at the shell prompt.</li> <li>2. To give understanding of various Python data types and expressions to solve relative problems</li> <li>3. To learn the use of Dictionary&amp; related functions</li> </ol>						
<b>Course Outcomes (COs):</b>						
<ol style="list-style-type: none"> <li>1. To utilize high-level data types such as lists and dictionaries</li> <li>2. To import and utilize a module • read from and write to a text file.</li> <li>3. understand the difference between mutable and immutable types</li> <li>4. To demonstration of IDE's: IDLE, IPython, IPython Notebook, hosted environments.</li> <li>5. To utilize high-level data types such as lists and dictionaries</li> </ol>						
<b>List of Experiments</b>						
<ol style="list-style-type: none"> <li>1. Demonstrate the working of 'id' and 'type' functions</li> <li>2. To find all prime numbers within a given range.</li> <li>3. To print 'n terms of Fibonacci series using iteration.</li> <li>4. To demonstrate use of slicing in string</li> <li>5. To add 'ing' at the end of a given string (length should be at least 3).</li> <li>6. To compute the frequency of the words from the input. The output should output after sorting the key alphanumerically</li> <li>7. Write a program that accepts a sequence of whitespace separated words as input and prints the words after removing all duplicate words and sorting them alphanumerically.</li> <li>8. To demonstrate use of list &amp; related functions</li> <li>9. To demonstrate use of Dictionary&amp; related functions</li> <li>10. To demonstrate use of tuple, set&amp; related functions</li> <li>11. To implement stack using list</li> <li>12. To implement queue using list</li> <li>13. To read and write from a file</li> <li>14. To copy a file</li> <li>15. To demonstrate working of classes and objects</li> <li>16. To demonstrate class method &amp; static method</li> <li>17. To demonstrate constructors</li> <li>18. To demonstrate inheritance</li> <li>19. To demonstrate aggregation/composition</li> <li>20. During the course student must be do project on:</li> <li>21. To create a small GUI application for insert, update and delete in a table using Oracle as</li> </ol>						

backend and front end for creating form

- 22.** Dice Rolling Simulator (This beginner-level project will help build a strong foundation for fundamental programming concepts)
- 23.** Number Guessing (To compute the difference between the two, and to check whether an actual number was inputted or not)
- 24.** Random Password Generator (Student can build a program that intakes some words from the user and then generates a random password using those words.
- 25.** At least one Project is mandatory for each student.

**Semester III**  
**Android App Development Lab**

Name of the Department- Computer Science & Engineering							
Course Name	Mobile App Development Lab	L		T		P	
Course Code		0		0		4	
Type of Course (use tick mark)		Core (✓)		PE()		OE ()	
Pre-requisite (if any)		Frequency (use tick marks)		Even ()		Odd (✓)	
				Either Sem ()		Every Sem ()	
<b>Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>							
<b>Lectures = 0</b>		<b>Tutorials = 0</b>		<b>Practical = 24</b>			
<b>Course Description:</b> Covers introductory mobile application development for the Android Operating System using XML and Java. Includes developing simple applications that could run on Android phones and tablets.							
<b>Learning objectives:</b>							
<ol style="list-style-type: none"> <li>1. Develop basic Android application</li> <li>2. Creating Activities</li> <li>3. Using Intents for activity communication</li> <li>4. Develop the GUI application.</li> </ol>							
<b>Course Outcomes (COs):</b>							
1. Understand android application hierarchy, UI components and their purpose							
2. Create activity, do activity to activity communication using intents and transfer data between/among intents.							
3. Apply style to android UI components							
4. Able to use and implement menus, notifications & implement notification using Notification.							
5. Configure and implement context menu and option menu as a part of android app.							
<b>List of Experiments</b>							
<b>11. List of Experiments</b>							
<ol style="list-style-type: none"> <li>1. Create a basic mobile application</li> <li>2. Working with forms</li> <li>3. Android App- working with intents</li> <li>4. Apply style and theme in an android app</li> <li>5. Create an Android app that does payment process via a context menu</li> <li>6. Create an Android app that does a currency converter operations using an options menu</li> <li>7. Create an Android notification app that displays notification about the messages received</li> <li>8. Create an Android app for sending data from first activity to second activity.</li> <li>9. Create an Android app for getting result from second activity (Using startActivityForResult)</li> <li>10. Create an Android app for storing user data using SQLITE</li> </ol>							

**Semester IV**  
**Embedded systems in Medicine**

Name of the Department- Computer Science & Engineering						
Course Name	Embedded systems in Medicine	L	T		P	
Course Code		3	0		0	
Type of Course (use tick mark)		Core (✓)	PE()		OE ()	
Pre-requisite (if any)	None	Frequency (use tick marks)	Even (✓)	Odd ()	Either Sem ()	Every Sem ()
<b>Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
Lectures = 36		Tutorials = 0	Practical = 0			
<b>Course Description</b>						
This course emphasizes on comprehensive treatment of embedded hardware and real time operating systems along with case studies, in tune with the requirements of Industry. The objective of this course is to enable the students to understand embedded-system programming and apply that knowledge to design and develop embedded solutions.						
<b>Learning objectives:</b>						
<ol style="list-style-type: none"> <li>1. Basic concepts of Embedded Systems</li> <li>2. Various techniques used for designing an embedded system.</li> <li>3. Real time system with an examples</li> </ol>						
<b>Course Outcomes (COs): At the end of this course, students will be able to</b>						
<ol style="list-style-type: none"> <li>1. Discuss the basics of embedded systems and its hardware units</li> <li>2. Identify the various tools and development process of embedded system</li> <li>3. Demonstrate the various I/O interfacing with microcontroller</li> <li>4. Create the programming for embedded system design</li> <li>5. Summarize the real time models, languages and operating systems</li> <li>6. Design a real time embedded system for biomedical applications</li> </ol>						
<b>Unit wise detailed content</b>						
<b>Unit-1</b>	<b>Number of lectures = 10</b>					
Definition and Classification : Overview of Processors and hardware units in an embedded system – Software embedded into the system – Exemplary Embedded Systems – Embedded Systems on a Chip (SoC) and the use of VLSI designed circuits - Embedded Hardware Architecture, Communication Interface Standards, Embedded System Development Process, Embedded Operating systems, Types of Embedded Operating systems.						
<b>Unit – 2</b>	<b>Number of lectures = 08</b>					
Intel MCS51 Architecture: Derivatives - Special Function Registers (SFR), I/O pins, ports and circuits, Instruction set, Addressing Modes, Assembly Language Programming, Timer and Counter Programming, Serial Communication, Connection to RS 232, Interrupts Programming, External Memory interfacing , Introduction to 16 bit Microcontroller. Interfacing of 8051 with ADC, sensors, stepper motor, key board, & DAC. PIC Microcontroller : Introduction, CPU architecture, registers, instruction sets addressing modes Loop timing, timers, Interrupts, Interrupt timing, I/o Expansion, I 2C Bus Operation Serial EEPROM, Analog to digital converter, UART-Baud Rate-Data Handling-Initialisation, Special Features - serial Programming-Parallel Slave Port .						
<b>Unit – 3</b>	<b>Number of</b>					

	<b>lectures = 10</b>	
<p>Embedded system evolution trends: Round - Robin, robin with Interrupts, function-One-Scheduling Architecture, Algorithms. Introduction to-assembler-compiler-cross compilers and Intergrated Development Environment (IDE). Object Oriented Interfacing, Recursion, Debugging strategies, Simulators. Task and Task States, tasks and data, semaphores and shared Data Operating system Services-Message queues-Timer Function- Events-Memory Management, Interrupt Routines in an RTOS environment, basic design Using RTOS.</p>		
<b>Unit – 4</b>	<b>Number of lectures = 08</b>	
<p>Applications: Real-Time Embedded Software Development, Sending a Message over a Serial Link, Simulation of a Process Control System, Controlling an Appliance from the RT Linux System, Embedded Database Applications, Embedded medical applications: Ophthalmology - Glaucoma screening device, Medical Imaging Acquisition User Interface, Drug delivery systems, Patient monitoring Systems.</p>		
<p>Brief Description of self-learning / E-learning component  The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University.  The link to the E-Learning portal.</p>		
<b>Books Recommended</b>		
<p>Text Books  1)Rushmer, " Medical Engineering", Academic Press  2)Yukihito Nose, " The Artificial Kidney", C V Mosby Co  3)Kennedy &amp; Blackie, " Electromedical Engineering"  4)Webstar, " Electronic Devices for Rehabilitation"</p>		
<b>Reference Books</b>		
<p>1)Myers, " Engineering in Heart and Blood Vessels", Wiley International  2) David Cooney, " Advanced in Bio medical Engineering", Marcel Decker Publications  3) David Cooney , "Biomedical Engineering Principles", Marcel Decker Publications  4) Ibrall &amp; Guyton , "Regulations and Control in Physiological Systems ", Instruments Society USA  5) Brown &amp; Gann, " Engineering in Physiology Vol 1 &amp; Vol 2 "</p>		

<b>List of Department Electives</b>					
<b>Specialization</b>	<b>Cyber Security &amp; Forensics</b>	<b>AIML</b>	<b>IoT</b>	<b>Blockchain</b>	<b>Data Analytics</b>
<b>DE-I</b>	Cyber Forensics and investigation	Neural Network and Deep learning(Vision and NLP)	Design of Smart Systems	Blockchain and Distributed Ledger Technology	Programming for Data Science
<b>DE-II</b>	Mobile Application Security & Penetration Testing	Data Science tools and techniques	Architecting Smart IoT Devices	Design and Development of Blockchain Applications	Big Data Technologies
<b>DE-III</b>	Cloud Security Essentials	Data Visualization	Application of IoT in Robotics	Programming Fundamentals : Golang and Solidity	Mining Massive Data

# **Cyber Security & Forensics**

## Cyber Forensics and Investigation

<b>1. Name of the Department:- Computer Science Engineering</b>						
<b>2. Course Name</b>	<b>Cyber Forensics and investigation</b>	<b>L</b>	<b>T</b>		<b>P</b>	
<b>3. Course Code</b>		<b>3</b>	<b>0</b>		<b>2</b>	
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE(✓)</b>		<b>OE ( )</b>	
<b>5. Pre-requisite (if any)</b>	C	<b>6. Frequency (use tick marks)</b>	Even ( )	Odd (✓)	Either Sem ( )	Every Sem ( )
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 24</b>		
<b>8. Course Description</b>						
This course enables the students to gain in-depth knowledge in the field of Computer forensics & Cyber Crime						
<b>9. Learning Objectives:</b>						
<ul style="list-style-type: none"> <li>I. To impart the basic concepts of Cyber Forensics.</li> <li>II. To understand different types of cyber attacks.</li> <li>III. To understand analysis of data to identify evidence, Technical Aspects &amp; Legal Aspects related to cyber crime.</li> </ul>						
<b>10. Course Outcomes (COs):</b>						
<ul style="list-style-type: none"> <li>I. Understand the fundamentals of Computer Forensics</li> <li>II. Learn the issues of Data Acquisition and Data Recovery</li> <li>III. Explore networking in cyber forensics</li> <li>IV. To learn, analyze and validate Forensics Data</li> <li>V. To study the tools and tactics associated with Cyber Forensics</li> </ul>						
<b>11. Unit wise detailed content</b>						
<b>Unit-1</b>	<b>Number of lectures = 08</b>					
Introduction to Cyber forensics: Information Security Investigations, Corporate Cyber Forensics, Scientific method in forensic analysis, investigating large scale Data breach cases. Analyzing Malicious software. Types of Computer Forensics Technology, Types of Military Computer Forensic Technology, Types of Law Enforcement: Computer Forensic Technology, Types of Business Computer Forensic Technology, Specialized Forensics Techniques, Hidden Data and How to Find It, Spyware and Adware, Encryption Methods and Vulnerabilities, Protecting Data from Being Compromised Internet Tracing Methods, Security and Wireless Technologies, Avoiding Pitfalls with Firewalls Biometric Security Systems.						
<b>Unit – 2</b>	<b>Number of lectures = 10</b>					
Types of Computer Forensics Systems: Internet Security Systems, Intrusion Detection Systems, Firewall Security Systems, Storage Area Network Security Systems, Network Disaster Recovery Systems, Public Key Infrastructure Systems, Wireless Network Security Systems, Satellite Encryption Security Systems, Instant Messaging (IM) Security Systems, Net Privacy Systems, Identity Management Security Systems, Identity Theft, Biometric Security Systems.						
<b>Unit – 3</b>	<b>Number of lectures = 08</b>					



Windows Forensic Analysis: Window artifacts, Evidence volatility, System time, Logged on user(s), Open files, MRUs, Network information, Process information, Service information, Windows Registry, Start up tasks, Memory dumping; Document Forensics: PDF structure, PDF analysis, MS Office Document structure and analysis, Macros, Windows thumbnails, Android Thumbnails.

**Unit – 4**

**Number of  
lectures =  
10**

Forensic Tools and Processing of Electronic Evidence: Introduction to Forensic Tools, Usage of Slack space, tools for Disk Imaging, Data Recovery, Vulnerability Assessment Tools, Encase and FTK tools, Anti Forensics and probable counters, retrieving information, process of computer forensics and digital investigations, processing of digital evidence, digital images, damaged SIM and data recovery, multimedia evidence, retrieving deleted data: desktops, laptops and mobiles, retrieving data from slack space, renamed file, ghosting, compressed files.

**12. Brief Description of self-learning / E-learning component**

The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University.

**13. Books Recommended**

**Text books:**

- I. John R. Vacca, Computer Forensics: Computer Crime Scene Investigation, 2nd Edition, Charles River Media, 2005

**Reference books:**

- II. Christof Paar, Jan Pelzl, Understanding Cryptography: A Textbook for Students and Practitioners, 2 nd Edition, Springer's, 2010
- III. Ali Jahangiri, Live Hacking: The Ultimate Guide to Hacking Techniques & Countermeasures for Ethical Hackers & IT Security Experts, Ali Jahangiri, 2009
- IV. Computer Forensics: Investigating Network Intrusions and Cyber Crime (Ec-Council Press Series: Computer Forensics), 2010
- V. Guide to Computer Forensics And Investigations Nelson, Bill ; Phillips, Amelia; Enfinger, Frank; Steuat, Christopher Thomson Course Technology.
- VI. Computer Forensics – Computer Crime Scene Investigation. Vacca, John R. Charles RiverMedia

## Cyber Forensics and investigation Lab

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	Cyber Forensics and investigation Lab	L	T	P		
<b>3. Course Code</b>		3	0	2		
<b>4. Type of Course (use tick mark)</b>		Core (✓)	PE(✓)	OE ()		
<b>5. Pre-requisite (if any)</b>	Computer Basics	<b>6. Frequency (use tick marks)</b>	Even ()	Odd (✓)	Either Sem()	Every Sem ()
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
Lectures = 0		Tutorials = 0		Practical = 10		
<b>8. Course Description</b>						
This course will cover the fundamentals of computer forensics and investigations. Topics will include historical and current computer forensic and investigative security issues						
<b>9. Learning Objectives:</b>						
<ol style="list-style-type: none"> <li>1. Explain how to conduct a digital forensics investigation.</li> <li>2. Report findings from digital forensic investigations.</li> <li>3. Perform recovery of digital evidence from various digital devices using a variety of software utilities.</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
The students will be able to:-						
<ol style="list-style-type: none"> <li>1. Utilize a systematic approach to computer investigations.</li> <li>2. Utilize various forensic tools to collect digital evidence.</li> <li>3. Perform digital forensics analysis upon Windows, MAC and LINUX operating systems</li> <li>4. Perform email investigations.</li> </ol>						
<b>11. Lab Experiments</b>						
Sr.No	Topics				CO Cover	
1	Study of Computer Forensics and different tools used for forensic investigation					
2	How to Recover Deleted Files using Forensics Tools					
3	Study the steps for hiding and extract any text file behind an image file/ Audio file using Command Prompt.				2	
4	How to Extract Exchangeable image file format (EXIF) Data from Image Files using Exifreader Software				1,3	
5	How to make the forensic image of the hard drive using EnCase Forensics.					
6	How to Restoring the Evidence Image using EnCase Forensics					
7	How to Collect Email Evidence in Victim PC					
8	How to Extracting Browser Artifacts					
9	How to View Last Activity of Your PC					
10	Find Last Connected USB on your system (USB Forensics)					
11	Comparison of two Files for forensics investigation by Compare IT software					
12	Live Forensics Case Investigation using Autopsy					

## Mobile Application Security & Penetration Testing

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	Mobile Application Security & Penetration Testing	<b>L</b>	<b>T</b>	<b>P</b>		
<b>3. Course Code</b>		<b>3</b>	<b>0</b>	<b>2</b>		
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE(✓)</b>	<b>OE ( )</b>		
<b>5. Pre-requisite (if any)</b>	Basic Java/IOS programming skills.	<b>6. Frequency (use tick marks)</b>	Even ( )	Odd (✓)	Either Sem( )	Every Sem ( )
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 0</b>		
<b>8. Course Description</b>						
Course benefits the career of Penetration Testers and IT security personnel in charge of defending their organization's applications and data.						
<b>10. Learning Objectives:</b>						
<ol style="list-style-type: none"> <li>1. To understand the different types of vulnerabilities that affect mobile applications and have the practical knowledge to attack and exploit them.</li> <li>2. To perform real world attacks on Android Devices and Apps.</li> <li>3. To learn How to Fuzz mobile apps.</li> <li>4. To learn Mobile applications reverse engineering.</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
<p style="text-align: center;">The students will be able to:-</p> <ol style="list-style-type: none"> <li>1. Perform real world attacks on Android Devices and Apps.</li> <li>2. Learn Mobile applications reverse engineering.</li> <li>3. Perform Penetration tests of mobile applications.</li> </ol>						
<b>11. Unit wise detailed content</b>						
<b>Unit-1</b>	<b>Number of lectures = 9</b>					
<b>ANDROID PENTESTING:</b> Android Architecture, Setting up a Test Environment , Android Build Process , Reversing APKs , Device Rooting , Android Application Fundamentals , Network Traffic , Device and Data Security , Tapjacking , Static Code Analysis , Dynamic Code Analysis						
<b>Unit – 2</b>	<b>Number of lectures = 9</b>					
<b>iOS PENTESTING:</b> iOS Architecture, Device Jailbreaking, Setting up a Testing Environment, iOS Building Process, Reversing iOS Apps, iOS Application Fundamentals, iOS Testing Fundamentals, Network Traffic, Device Administrator, Dynamic Analysis						
<b>Unit – 3</b>	<b>Number of lectures = 9</b>					
<p>. <b>Reversing APKs:</b> APKTool , Dex2Jar , JD-GUI , Smali/Backsmali , Obfuscation , Additional APK Contents , Hardware Optimization , OEM Apps</p> <p>. <b>Device Rooting:</b> What is Rooting , SuperUser and SuperSU, . Potential Issues , Custom ROMs , OmniROM and CyanogenMod, Google Nexus, Implication of Rooting , Rooting for Testing</p>						

<b>Unit – 4</b>	<b>Number of lectures = 9</b>	
<p><b>Device and Data Security:</b> Data Storage, Internal Storage, External Storage. Device Administration API ,MDM Solutions , Root Detection , Third-Party Code , SDK ,Libraries , Device Tracking</p> <p><b>Static Code Analysis :</b>Static Code Analysis, Vulnerable Code Snippet, . Vulnerability Exploitation , SQL Injection , Selection query, Direct Using User Input , Partial Parametrization, Full Parametrization,ContentProviders, ContentResolver , Path/Directory Traversal, Vulnerable Activities , android permission , intentMessage, Vulnerable Receivers , Vulnerable Services ,Shared Preferences , Local Databases, . Sqlite3, .Tools :. Drozer , QARK</p>		
<p><b>12. Brief Description of self-learning / E-learning component</b></p> <p>The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University.</p> <p>The link to the E-Learning portal.  <a href="https://elearning.sgtuniversity.ac.in/course-category/">https://elearning.sgtuniversity.ac.in/course-category/</a></p>		
<p><b>13. Books Recommended</b></p>		
<p><b>Text Books</b></p> <p>Mobile Application Penetration Testing, <b>Vijay Kumar Velu</b> ,2016.</p> <ul style="list-style-type: none"> <li>•</li> </ul>		
<p><b>14. Reference Books</b></p>		
<p>“The Pentester Blueprint” by Phillip J. Wylie and Kim Crawley, Wiley 2021.</p> <p>“Penetration Testing For Dummies” by Robert Shimonski, 2020</p> <p>“AWS Penetration Testing” by Jonathan Helmus,2019.</p>		

## Mobile Application Security & Penetration Testing Lab

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	Mobile Application Security & Penetration TestingLab	<b>L</b>	<b>T</b>		<b>P</b>	
<b>3. Course Code</b>		<b>3</b>	<b>0</b>		<b>2</b>	
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE(✓)</b>		<b>OE ()</b>	
<b>5. Pre-requisite (if any)</b>	Computer Basics	<b>6. Frequency (use tick marks)</b>	Even ()	Odd (✓)	Either Sem()	Every Sem ()
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 0</b>		<b>Tutorials = 0</b>		<b>Practical = 10</b>		
<b>8. Course Description</b>						
This course will walk you through the process of identifying security issues on Android and iOS applications, using a wide variety of techniques including Reverse Engineering, Static/Dynamic/Runtime and Network Analysis.						
<b>11. Learning Objectives:</b>						
<ol style="list-style-type: none"> <li>1. Teaches students mobile application programming.</li> <li>2. Teaches you how to jailbreak or root iOS/Android devices.</li> <li>3. Teaches give you a certification without any effort.</li> <li>4. You can memorize to pass a multiple-choice test.</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
The students will be able to:-						
<ol style="list-style-type: none"> <li>1. To acquaint students with the practical aspects of Design.</li> <li>2. To understand the importance of User engagement and Experience.</li> <li>3. To learn various development techniques</li> </ol>						
<b>11. Lab Experiments</b>						
Sr.No	Topics			CO		
1	To implement Device Rooting.					
2	To implement Tapjacking.					
3	To implement Android Virtual Machine, Dalvik Executable (DEX), Optimized DEX (ODEX) , Android NDK.			2		
4	To learn Using Emulators, AVD Manager, Create Virtual Device, System Images, Start the emulator.			1,3		
5	To implement Reversing APKs 4.1. , APKTool 4.2. , Dex2Jar.					
6	To implement Su, SuperUser and SuperSU rooting.					
7	To implement Proxy Configuration. Burp Suite, CA Certificates.					
8	To learn Device Administration API, MDM Solutions,. Root Detection.					
9	To implement SQL Injection. Selection query, Direct Using User Input.					
10	To implement Vulnerable Activities.					

## Cloud Security Essentials

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	<b>Cloud Security Essentials</b>	<b>L</b>	<b>T</b>	<b>P</b>		
<b>3. Course Code</b>		<b>3</b>	<b>0</b>	<b>2</b>		
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE(✓)</b>		<b>OE ()</b>	
<b>5. Pre-requisite (if any)</b>		<b>6. Frequency (use tick marks)</b>	Even (✓)	Odd ()	Either Sem()	Every Sem ()
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>	<b>Practical = 0</b>			
<b>8. Course Description</b>						
<p><b>In cloud computing, we can manipulate, configure and access the hardware and software remotely. In general, cloud computing is accessing and storing the files and databases over the internet instead of accessing it on your computer's hard drive. Cloud computing offers platform independence, the software is not required to be installed on any PC. There is portability in cloud computing.</b></p>						
<b>12. Learning Objectives:</b>						
<ol style="list-style-type: none"> <li>1. Understand the computing paradigm and cloud computing</li> <li>2. Understand the architecture of cloud computing</li> <li>3. Understand and use the service models and deployments</li> <li>4. Work on any real cloud service</li> <li>5. Understand the service management and security of cloud</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
<p>The students will be able to:-</p> <ol style="list-style-type: none"> <li>1. Clarify the different definitions of cloud computing and its characteristics.</li> <li>2. Explain the principles of hardware virtualization and its importance for cloud computing.</li> <li>3. Motivate the importance of data centers for clouds, and explain how to design and construct a data center.</li> </ol>						
<b>11. Unit wise detailed content</b>						
<b>Unit-1</b>	<b>Number of lectures = 9</b>					
<p><b>INTRODUCTION</b> Overview of computing paradigms, Recent trends in computing, evolution of cloud computing, Overview of cloud computing, Cloud computing-Concepts, properties, characteristics, Role of open standards. Cloud computing architecture, Cloud service delivery models (XAAS), Cloud Deployment models</p>						
<b>Unit – 2</b>	<b>Number of lectures = 9</b>					
<p><b>INFRASTRUCTURE AS A SERVICE</b> Introduction, Hypervisors, Resource virtualization, Examples, How to implement IAAS  <b>PLATFORM AS A SERVICE</b> Introduction, Cloud Platform and Management, Examples, How to implement PAAS  <b>SOFTWARE AS A SERVICE</b> Introduction, Web services, Web 2.0, Web OS, Examples, How to implement SAAS</p>						
<b>Unit – 3</b>	<b>Number of lectures = 9</b>					
<p><b>SERVICE MANAGEMENT IN CLOUD COMPUTING</b> Service Orchestration -Cloud computing and Service Management, Service Level Agreements (SLAs), Billing &amp; Accounting, Comparing scaling hardware, economics of scaling, managing data. Cloud performance, Existing project experience</p>						

<b>Unit – 4</b>	<b>Number of lectures = 9</b>	
<p><b>CLOUD SECURITY</b> Infrastructure security, Data Security, Storage Identity and Access Management, Access Control, Trust and Reputation, Authentication in Cloud computing,  <b>CASE STUDY ON OPEN SOURCE AND REAL CLOUD SERVICES</b> Eucalyptus, VMware Cloud, IBM Bluemix, Google Cloud services, Amazon Web services</p>		
<p><b>12. Brief Description of self-learning / E-learning component</b>  The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University.  The link to the E-Learning portal.  <a href="https://elearning.sgtuniversity.ac.in/course-category/">https://elearning.sgtuniversity.ac.in/course-category/</a></p>		
<p><b>13. Books Recommended</b></p>		
<p><b>Text Books</b></p> <ul style="list-style-type: none"> <li>• . Barrie Sosinsky: "Cloud Computing Bible", Wiley-India, 2010</li> </ul>		
<p><b>14. Reference Books</b></p>		
<ul style="list-style-type: none"> <li>• . RajkumarBuyya, James Broberg, Andrzej M. Goscinski: "Cloud Computing: Principles and Paradigms", Wiley, 2011</li> <li>• Nikos Antonopoulos, Lee Gillam: "Cloud Computing: Principles, Systems and Applications", Springer, 2012</li> <li>• Ronald L. Krutz, Russell Dean Vines: "Cloud Security: A Comprehensive Guide to Secure Cloud Computing", Wiley-India, 2010</li> <li>• Tim Mather, Subra Kumara swamy, ShahedLatif, Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance, O'Reilly Media, 2009.</li> </ul>		

## Cloud Security Essentials Lab

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	<b>Cloud Security Essentials Lab</b>	<b>L</b>	<b>T</b>		<b>P</b>	
<b>3. Course Code</b>		<b>3</b>	<b>0</b>		<b>2</b>	
<b>4. Type of Course (use tick mark)</b>		<b>Core (☐ )</b>	<b>PE(☐)</b>		<b>OE ( )</b>	
<b>5. Pre-requisite (if any)</b>	iOS Basics	<b>6. Frequency (use tick marks)</b>	Even (☐)	Odd ( )	Either Sem.( )	Every Sem. ( )
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 0</b>		<b>Tutorials = 0</b>		<b>Practical = 10</b>		
<b>8. Course Description</b>						
This course will walk you through the process of identifying security issues on Android and iOS applications, using a wide variety of techniques including Reverse Engineering, Static/Dynamic/Runtime and Network Analysis.						
<b>13. Learning Objectives:</b>						
<ol style="list-style-type: none"> <li>1. Teaches students Cloud programming.</li> <li>2. Teaches you how to implement cloud devices.</li> <li>3. Teaches give you a knowledge of cloud infrastructure.</li> <li>4. You can memorize to pass a multiple-choice test.</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
The students will be able to:-						
<ol style="list-style-type: none"> <li>1. To acquaint students with the practical aspects of Design.</li> <li>2. To understand the importance of User engagement and Experience.</li> <li>3. To learn various development technique</li> </ol>						
<b>11. Lab Experiments</b>						

Sr. No.	Title	CO
1	To Identify users and manage their access.	
2	Learn to Protect applications and data, with appropriate security controls across network, data, and application access	1,2
3	Learn to Gain visibility and insights into security, compliance, and threat posture.	
4	To Inject security-based principles into the development and operation of cloud-based services.	1,3
5	To learn to design IaaS,PaaS and SaaS architecture..	
6	To design a Cloud Ecosystem.	
7	To learn Workload protection, integrated with DevSecOps	
8	To implement security posture and compliance management	
9	To implement Threat management	



**AIML**

## Neural Network And Deep Learning

Name of the Department- Computer Science & Engineering						
2. Course Name	<b>Neural Network And Deep Learning</b>	L	T	P		
3. Course Code		3	0	2		
4. Type of Course (use tick mark)		Core (✓)	PE(✓)	OE ( )		
5. Pre-requisite (if any)	NIL	6. Frequency (use tick marks)	Even ( )	Odd (✓)	Either Sem ( )	Every Sem ( )
7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)						
Lectures = 36		Tutorials = 0	Practical = 0			
8. Course Description						
A convolutional deep learning neural network is built using Keras to show how deep learning is used in specialized neural networks. This course provides the necessary required background to understand ROI's Time Series Analysis and Natural Language Processing courses.						
9. Learning objectives:						
<ol style="list-style-type: none"> <li>1. Understand the context of neural networks and deep learning</li> <li>2. Know how to use a neural network</li> <li>3. Understand the data needs of deep learning</li> <li>4. Have a working knowledge of neural networks and deep learning</li> <li>5. Explore the parameters for neural networks</li> </ol>						
10. Course Outcomes (COs):						
<ol style="list-style-type: none"> <li>1. To introduce the fundamental techniques and principles of Neural Networks</li> <li>2. To study the different models in ANN and their applications</li> <li>3. To familiarize deep learning concepts with Convolutional Neural Network case studies</li> </ol>						
11. Unit wise detailed content						
Unit-1	Number of lectures = 10					
<p><b>The neural network:</b> The neuron, linear perceptron, feed-forward neural network, limitations of linear neurons, sigmoid, tanh, relu neurons, softmax output layer, information theory, cross entropy, Kullback-Leibler divergence</p> <p><b>Training feed-forward neural network:</b> Gradient Descent, delta rules and learning rates, gradient descent with sigmoidal neurons, the backpropagation algorithms, stochastic and minibatch gradient descent, test sets, validation sets and overfitting, preventing overfitting</p>						
Unit – 2	Number of lectures = 9					
<p><b>TensorFlow:</b> Computation graphs, graphs, sessions and fetches, constructing and managing graph, flowing tensors, sessions, data types, tensor arrays and shapes, names, variables, placeholders and simple optimization, linear regression and logistic regression using tensorflow</p> <p><b>Implement Neural Network:</b> Introduction to Keras, Build neural network using Keras, Evaluating models, data preprocessing, feature engineering, feature learning, overfitting, underfitting, weight regularization, dropout, universal workflow of deep learning</p>						
Unit – 3	Number of lectures = 8					
<p><b>Moving beyond gradient descent:</b> Local minima vs global minima vs saddle, model identifiability, correcting gradient points in wrong directions, Momentum based optimization,</p>						

<p>second order methods, learning rate adaption, adagrad, rmsprop, adam</p> <p><b>Convolutional Neural Network:</b> Convolution operation, filters and feature maps, motivation, sparse interactions, parameter sharing and equivariant representation, padding and stride, max pooling, full architectural description of convolutional network, build cnn using data augmentation, using pretrained convnet, visualize what convnet learn.</p>		
Unit – 4	Number of lectures =	
	9	
<p><b>Embedding and Representation Learning:</b> Principle component analysis, working with text data, one-hot encoding of words and characters, word embedding, autoencoder architecture, denoising, sparsity, Word2vec framework, Skip-Gram architecture.</p>		
<p><b>Models for Sequence Analysis:</b> Analysing Variable-length inputs, Seq2seq with neural n-gram, part of speech tagger, dependency parse, syntaxnet, recurrent neural network, challenges with vanishing gradients, long short term memory units</p>		
<p>13. Books Recommended</p>		
<p>Text Books</p> <ol style="list-style-type: none"> <li>1. CharuC.Aggarwal “Neural Networks and Deep learning” Springer International Publishing, 2018</li> <li>2. Satish Kumar, “Neural Networks, A Classroom Approach”, Tata McGraw -Hill, 2007.</li> <li>3. Simon Haykin, “Neural Networks, A Comprehensive Foundation”, 2nd Edition, Addison Wesley Longman, 2001.</li> </ol>		
<p>Reference Books</p> <ol style="list-style-type: none"> <li>1. Bishop, Christopher M. Pattern Recognition and Machine Learning. Springer, 2006</li> <li>2. Duda, Richard, Peter Hart, and David Stork. Pattern Classification. 2nd ed. New York, NY: Wiley-Interscience, 2000.</li> </ol>		

## Neural Network and Deep learning Lab

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	<b>Neural Network and Deep learning Lab</b>	<b>L</b>	<b>T</b>	<b>P</b>		
<b>3. Course Code</b>		<b>3</b>	<b>0</b>	<b>2</b>		
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE(✓)</b>		<b>OE ()</b>	
<b>5. Pre-requisite (if any)</b>		<b>6. Frequency (use tick marks)</b>	Even ()	Odd (✓)	Either Sem()	Every Sem ()
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 0</b>		<b>Tutorials = 0</b>		<b>Practical = 10</b>		
<b>8. Course Description</b>						
This course Introduce major deep learning algorithms, the problem settings, and their applications to solve real world problems.						
<b>9.Learning Objectives:</b>						
<ol style="list-style-type: none"> <li>1. to provide an introduction to the field of artificial neural networks and machine learning;</li> <li>2. to teach students how to solve practical problems via implementation of these techniques via simulation;</li> <li>3. to promote further independent learning on the topics of artificial neural networks and machine learning;</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
The students will be able to:-						
<ol style="list-style-type: none"> <li>1. Understand the characteristics and types of artificial neural network and remember working of biological Neuron and Artificial Neural Network.</li> <li>2. Apply learning algorithms on perceptron and apply back propagation learning on Neural Network.</li> <li>3. Apply Feedback NN and plot a Boltzmann machine and associative memory on various application .</li> </ol>						
<b>11. Lab Experiments</b>						
<b>Sr.No</b>	<b>Topics</b>					<b>CO</b>
1	To Write a program to implement Perceptron.					
2	To write a program to implement AND OR gates using Perceptron					
3	To implement Crab Classification using pattern net					
4	To write a program to implement Wine Classification using Back propagation.					
5	To write a MatLab Script containing four functions Addition, Subtraction, Multiply and Divide functions					3
6	Write a program to implement classification of linearly separable Data with a perceptron					2,3
7	To study Long Short Term Memory for Time Series Prediction					
8	To study Convolutional Neural Network and Recurrent Neural Network					
9	To study ImageNet, GoogleNet, ResNet convolutional Neural Networks					
10	To study the use of Long Short Term Memory / Gated Recurrent Units to predict the stock prices based on historic data					1,2

## Data Science Tools And Techniques

Name of the Department- Computer Science & Engineering						
2. Course Name	Data Science Tools And Techniques	L	T	P		
3. Course Code		3	0	2		
4. Type of Course (use tick mark)		Core (✓)	PE(✓)	OE ()		
5. Pre-requisite (if any)	NIL	6. Frequency (use tick marks)	Even ( )	Odd (✓)	Either Sem ( )	Every Sem ( )
7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)						
Lectures = 36		Tutorials = 0		Practical = 0		
8. Course Description						
The student learns the architecture of HDFS and MapReduce along with other tools such as pig, hive, spark, Zookeeper, HBase						
9. Learning objectives:						
<ol style="list-style-type: none"> <li>1. The objective of this course is to teach students the conceptual framework of Big Data,</li> <li>2. Learn Virtualization, MapReduce, HDFS</li> <li>3. Learn Pig, Hive, Spark, ZooKeeper, HBase.</li> </ol>						
10. Course Outcomes (COs):						
On completion of this course, the students are expected to learn						
<ol style="list-style-type: none"> <li>1. Concepts of Hadoop and HDFS</li> <li>2. Concepts of Map Reduce</li> <li>3. Big data tools Pig, Hive, Spark, Zookeeper, HBase</li> </ol>						
11. Unit wise detailed content						
Unit-1	Number of lectures = 10					
<b>Big Data:</b> Fundamentals of Big Data, defining big data, building successful big data management architecture, big data journey <b>Big Data Types:</b> Structured and unstructured data types, real time and non-real time requirements <b>Distributed Computing:</b> History of distributed computing, basics of distributed computing						
Unit – 2	Number of lectures = 9					
<b>Big Data Technology Foundation:</b> Big Data stack, redundant physical infrastructure, security infrastructure, operational databases, organising data services and tools, analytical data warehouse, big data analytics <b>Virtualization:</b> Basics of virtualization, hypervisor, abstraction and virtualization, implementing virtualization with big data <b>Cloud and Big Data:</b> Defining cloud, cloud deployment and delivery models, cloud as an imperative for big data, use the cloud for big data						
Unit – 3	Number of lectures = 8					
<b>Operational Databases:</b> Relational database, nonrelational database, key-value pair databases, document databases, columnar databases, graph databases, spatial databases						

**MapReduce Fundamentals:** Origin of MapReduce, map function, reduce function, putting map and reduce together, optimizing map reduce  
**Hadoop:** Discovering Hadoop, Hadoop distributed file system, Hadoop MapReduce, Hadoop file system, dataflow, Hadoop I/O, data integrity, compression, serialization, file-based data structure

Unit – 4	Number of lectures = 9	
----------	---------------------------	--

**Avro:** Avro data types and schemas, in-memory serialization and deserialization, avro datafiles, schema resolution  
**Pig:** Comparison with databases, pig latin, user defined functions, data processing operators  
**Hive:** Running hive, comparison with traditional databases, HiveQL, tables, querying data, user-defined functions  
**Spark:** Resilient distributed datasets, shared variables, anatomy of a spark job run, executors and cluster managers,  
**HBase:** HBase basics, concepts, clients, HBase vs RDBMS, Praxis  
**ZooKeeper:** ZooKeeper services, building application with ZooKeeper

### 13. Books Recommended

#### Text Books

Hadoop: The Definitive Guide, 4th Edition by Tom White - Shroff Publishers & Distributers Private Limited - Mumbai; Fourth edition (2015)

#### Reference Books

Big Data: Principles and Best Practices of Scalable Real-time Data Systems by James Warren and Nathan Marz, Manning Publications (2015) On Writing Well. William Zinsser. Harper Resource Book. 2001

## Data Science tools and techniques Lab

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	Data Science tools and techniques Lab	L	T	P		
<b>3. Course Code</b>		3	0	2		
<b>4. Type of Course (use tick mark)</b>		Core (✓)	PE(✓)	OE ()		
<b>5. Pre-requisite (if any)</b>		<b>6. Frequency (use tick marks)</b>	Even ()	Odd (✓)	Either Sem()	Every Sem ()
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
Lectures = 0		Tutorials = 0		Practical = 10		
<b>8. Course Description</b>						
This course provides the data and study data modeling to draw useful information out of the data, which is conclusive and helps in decision making for a certain problem or query.						
<b>Learning Objectives:</b>						
<ol style="list-style-type: none"> <li>1. Understand the R Programming Language.</li> <li>2. Exposure on Solving of data science problems.</li> <li>3. Understand The classification and Regression Model.</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
The students will be able to:-						
<ol style="list-style-type: none"> <li>1. Understand the features of R Programming Language.</li> <li>2. Understanding statistical techniques in R</li> <li>3. Understand The Regression Model.</li> </ol>						
<b>11. Lab Experiments</b>						

Sr.No	Topics	CO
1	R AS CALCULATOR APPLICATION a. Using with and without R objects on console b. Using mathematical functions on console c. Write an R script, to create R objects for calculator application and save in a specified location in disk.	1
2	DESCRIPTIVE STATISTICS IN R a. Write an R script to find basic descriptive statistics using summary, str, quartile function on mtcars& cars datasets. b. Write an R script to find subset of dataset by using subset (), aggregate () functions on iris dataset.	1,2
3	READING AND WRITING DIFFERENT TYPES OF DATASETS a. Reading different types of data sets (.txt, .csv) from Web and disk and writing in file in specific disk location.	2

	<p>b. Reading Excel data sheet in R.  c. Reading XML dataset in R.</p>	
4	<p><b>VISUALIZATIONS</b>  a. Find the data distributions using box and scatter plot.  b. Find the outliers using plot.  c. Plot the histogram, bar chart and pie chart on sample data.</p>	1,3
5	<p><b>CORRELATION AND COVARIANCE</b>  a. Find the correlation matrix.  b. Plot the correlation plot on dataset and visualize giving an overview of relationships among data on iris data.  c. Analysis of covariance: variance (ANOVA), if data have categorical variables on iris data.</p>	3
6	<p><b>REGRESSION MODEL</b>  Import a data from web storage. Name the dataset and now do Logistic Regression to find out relation between variables that are affecting the admission of a student in a institute based on his or her GRE score, GPA obtained and rank of the student. Also check the model is fit or not. Require (foreign), require (MASS)</p>	2,3
7	<p><b>MULTIPLE REGRESSION MODEL</b>  Apply multiple regressions, if data have a continuous Independent variable. Apply on above dataset.</p>	3
8	<p><b>REGRESSION MODEL FOR PREDICTION</b>  Apply regression Model techniques to predict the data on above dataset</p>	1
9	<p><b>CLASSIFICATION MODEL</b>  a. Install relevant package for classification.  b. Choose classifier for classification problem.  c. Evaluate the performance of classifier.</p>	2
10	<p><b>CLUSTERING MODEL</b> a. Clustering algorithms for unsupervised classification. b. Plot the cluster data using R visualizations</p>	1,2



## Data Visualization

Name of the Department- Computer Science & Engineering						
2. Course Name	<b>Data Visualization</b>	L	T	P		
3. Course Code		3	0	2		
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE(✓)</b>		<b>OE ( )</b>	
5. Pre-requisite (if any)	<b>NIL</b>	6. Frequency (use tick marks)	Even (✓)	Odd ( )	Either Sem ( )	Every Sem ( )
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 0</b>		
<b>8. Course Description</b>						
Data visualization techniques allow people to use their perception to better understand this data. The goal of this course is to introduce students to data visualization including both the principles and techniques. Students will learn the value of visualization, specific techniques in information visualization and scientific visualization, and how understand how to best leverage visualization methods.						
<b>9. Learning objectives:</b>						
1. To understand the various types of data, apply and evaluate the principles of data visualization. 2. Acquire skills to apply visualization techniques to a problem and its associated dataset. 3. To apply structured approach to create effective visualizations thereby building visualization dashboard to support decision making.						
<b>10. Course Outcomes (COs):</b>						
1. Identify the different data types, visualization types to bring out the insight. Relate the visualization towards the problem based on the dataset. 2. Identify the different attributes and showcasing them in plots. Identify and create various visualizations for geospatial and table data. 3. Ability to visualize categorical, quantitative and text data. Illustrate the integration of visualization tools with hadoop. 4. Ability to visualize categorical, quantitative and text data						
<b>11. Unit wise detailed content</b>						
<b>Unit-1</b>	<b>Number of lectures = 10</b>					
Overview of Data analysis, Introduction to Data visualization, Working with statistical formulas - Logical and financial functions , Data Validation & data models, Power Map for visualize data , Power BI-Business Intelligence , Data Analysis using statistical methods, Dashboard designing.						
<b>Unit – 2</b>	<b>Number of lectures = 9</b>					
Heat Map, Tree Map, Smart Chart,Azure Machine learning , Column Chart, Line Chart , Pie,Bar, Area, Scatter Chart, Data Series, Axes , Chart Sheet , Trendline , Error Bars, Sparklines, Combination Chart, Gauge, Thermometer Chart , Gantt Chart , Pareto Chart etc , Frequency Distribution, Pivot Chart, Slicers , Tables: Structured References, Table Styles , What-If Analysis: Data Tables, Goal Seek, Quadratic Equation , Transportation Problem, Maximum Flow Problem, Sensitivity Analysis, Histogram, Descriptive, Statistics, Anova, F-Test, t-Test, Moving, Average, Exponential Smoothing   Correlation model   Regression model						

<b>Unit – 3</b>	<b>Number of lectures = 8</b>	
<p>What is Tableau? What does the Tableau product suite comprise of? How Does Tableau Work? Tableau Architecture, What is My Tableau Repository? Connecting to Data &amp; Introduction to data source concepts, Understanding the Tableau workspace, Dimensions and Measures, Data Types &amp; Default Properties, Building basic views, Saving and Sharing your work-overview</p>		
<b>Unit – 4</b>	<b>Number of lectures = 9</b>	
<p>Date Aggregations and Date parts, Cross tab &amp; Tabular charts, Totals &amp; Subtotals, Bar Charts &amp; Stacked Bars, Trend lines, Forecasting, Filters, Context filters, Line Graphs with Date &amp; Without Date, Tree maps, Scatter Plots</p>		
13. Books Recommended		
Text Books		
<p><b>13. Books Recommended</b></p> <p>I. "Information Dashboard Design: Displaying Data for At-a-glance Monitoring" by Stephen Few Edition, 2014.</p> <p>2. Y. Daniel Liang, Introduction to Java programming-comprehensive version-Tenth Edition, Pearson ltd 2015.</p>		
Reference Books		
<p>1. "Beautiful Visualization, Looking at Data Through the Eyes of Experts by Julie Steele, Noah Iliinsky"</p> <p>2. "The Accidental Analyst: Show Your Data Who's Boss" by Eileen and Stephen McDaniel</p>		

## Data Visualization Lab

<b>1. Name of the Department- Computer Science &amp; Engineering</b>					
<b>2. Course Name</b>	<b>Data Visualization Lab</b>	<b>L</b>	<b>T</b>	<b>P</b>	
<b>3. Course Code</b>		<b>3</b>	<b>0</b>	<b>2</b>	
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE(✓)</b>		<b>OE ()</b>
<b>5. Pre-requisite (if any)</b>		<b>6. Frequency (use tick marks)</b>	Even (✓)	Odd ()	Either Sem() Every Sem ()
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>					
<b>Lectures = 0</b>		<b>Tutorials = 0</b>		<b>Practical = 10</b>	
<b>8. Course Description</b>					
This course will cover introduction to data visualization, where student will learn how to design, build, and evaluate visualizations for different types of data, disciplines, and domains					
<b>14. Learning Objectives:</b>					
<ol style="list-style-type: none"> <li>1. Explain how visualize the data..</li> <li>2. Explain how to use the Geospatial Data data.</li> <li>3. Explain how to use the color in good design.</li> </ol>					
<b>10. Course Outcomes (COs):</b>					
The students will be able to:-					
<ol style="list-style-type: none"> <li>1. Understand basics of Data Visualization</li> <li>2. Implement visualization of distributions.</li> <li>3. Write programs on visualization of time series, proportions &amp; associations</li> <li>4. Apply visualization on Trends and uncertainty</li> </ol>					
<b>11. Lab Experiments</b>					

Sr.No	Topics	CO Covere
1	Download the House Pricing dataset from Kaggle and map the values to Aesthetics	
2	Use different Color scales on the Rainfall Prediction dataset	
3	Create different Bar plots for variables in any dataset	
4	Show an example of Skewed data and removal of skewedness	
5	For a sales dataset do a Time Series visualization	
6	Build a Scatterplot and suggest dimension reduction	
7	Use Geospatial Data-Projections on datasets in <a href="http://www.gisinindia.com/directory/gis-data-for-india">http://www.gisinindia.com/directory/gis-data-for-india</a>	3
8	Create the a trend line with a confidence band in any suitable dataset	
9	Illustrate Partial Transparency and Jittering	
10	Illustrate usage of different color codes	

# **IoT**

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	<b>Design of Smart Systems</b>	<b>L</b>	<b>T</b>	<b>P</b>		
<b>3. Course Code</b>		<b>3</b>	<b>0</b>	<b>2</b>		
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE(✓)</b>	<b>OE ()</b>		
<b>5. Pre-requisite (if any)</b>		<b>6. Frequency (use tick marks)</b>	Even ( )	Odd (✓)	Either Sem()	Every Sem ()
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>	<b>Practical = 0</b>			
<b>8. Course Description</b>						
This course equips students with essential tools frequently used to impart intelligence to a variety of systems. Applications of these tools in the design and development of smart systems are illustrated.						
<b>15. Learning Objectives:</b>						
<ol style="list-style-type: none"> <li>To develop an understanding of Smart Systems, their implementation and applications.</li> <li>Interpret and explain the impact of Smart Systems, ethical, legal, social, environmental implications.</li> <li>Explain concepts used in Smart Systems and associated architectures.</li> <li>Explain the major Smart Systems application areas and techniques used within them</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
<p>Understand the basic principles behind smart systems.</p> <ol style="list-style-type: none"> <li>Understand the interfacing between the sensors and MCU.</li> <li>Understand the different control techniques.</li> <li>Understand the communication for smart sensors</li> </ol>						
<b>11. Unit wise detailed content</b>						
<b>Unit-1</b>	<b>Number of lectures = 9</b>					
INTRODUCTION TO SENSOR DEVICES : Piezoresistive pressure sensor- Piezoresistive Accelerometer - Capacitive Sensing- Accelerometer and Microphone - Resonant Sensor and Vibratory Gyroscope - Low-Power, Low Voltage Sensors- Micro Electro Mechanical Systems Analysis and Design of MEMS Devices- Nano Sensors						
<b>Unit – 2</b>	<b>Number of lectures = 9</b>					
INTERFACING SENSOR INFORMATION AND MCU: Amplification and Signal Conditioning- Integrated Signal Conditioning- Digital conversion- MCU ControlMCUs for Sensor Interface- Techniques and System Considerations- Sensor Integration.						
<b>Unit – 3</b>	<b>Number of lectures = 9</b>					
CONTROL TECHNIQUES AND STANDARDS : Control of Sensors using - State Machines, Fuzzy Logic, Neural Networks, Adaptive Control. Control Application using - CISC, RISC, DSP Control and IEEE 1451 Standards						
<b>1. Unit – 4</b>	<b>Number of lectures = 9</b>					
COMMUNICATION FOR SMART SENSORS:Wireless Data Communications- RF Sensing- Telemetry- Automotive Protocols- Industrial NetworksHome Automation- MCU Protocols. PACKAGING, TESTING AND RELIABILITY IMPLICATIONS OF SMART SENSORS: Semiconductor Packaging- Hybrid Packaging- Packaging for Monolithic Sensors- Reliability ImplicationsTesting Smart Sensors- HVAC Sensor Chip.						

**12. Brief Description of self-learning / E-learning component**

The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University.

The link to the E-Learning portal.

<https://elearning.sgtuniversity.ac.in/course-category/>

**13. Books Recommended****Text Books**

- Randy Frank, "Understanding Smart Sensors", Artech House, Second Edition, 2011 Boston,
- Minhang Bao, "Analysis and design principles of MEMS devices", Elsevier Publications, 2005, USA.
- Nadim Maluf and Kirt Williams, "An Introduction to Micro Electro Mechanical Systems Engineering", Second Edition, Artech House Publishers, June 2004, USA

**14. Reference Books**

- Gabriel M. Rebeiz, "RF MEMS: Theory, Design, and Technology", Wiley-Interscience; 1st edition, 2002, UK
- John A. Pelesko and David H. Bernstein, " Modeling MEMS and NEMS", CRC Press, 2002, UK
- Rai-choudhury, "MEMS and MOEMS Technology and Applications", PHI, 2010.

## Design of Smart Systems Lab

<b>1. Name of the Department- Computer Science &amp; Engineering</b>					
<b>2. Course Name</b>	Design of Smart Systems Lab	<b>L</b>	<b>T</b>	<b>P</b>	
<b>3. Course Code</b>		<b>3</b>	<b>0</b>	<b>2</b>	
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE(✓)</b>	<b>OE ( )</b>	
<b>5. Pre-requisite (if any)</b>		<b>6. Frequency (use tick marks)</b>	Even ( )	Odd (✓)	Either Sem( ) Every Sem ( )
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>					
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 0</b>	
<b>8. Course Description</b>					
This course introduces the basic of Arduino/Raspberry Pi. And programming on it.					
<b>16. Learning Objectives:</b>					
<ol style="list-style-type: none"> <li>1. To study IOT, their characteristics of components and basic awareness of Arduino/Raspberry Pi.</li> <li>2. To study various supporting OS platforms for Raspberry-Pi /Beagle board</li> <li>3. To study the fundamentals of connectivity schemes of Raspberry-Pi /Beagle board</li> <li>4. To facilitate the development and application of problem-solving skills in students.</li> </ol>					
<b>10. Course Outcomes (COs):</b>					
The students will be able to:-					
<ol style="list-style-type: none"> <li>1. understand IOT, Arduino/Raspberry Pi, and also able to install software setup of Arduino/Raspberry Pi.</li> <li>2. understand the different supporting OS platforms of Raspberry-Pi /Beagle board</li> <li>3. use Raspberry-Pi /Beagle board circuit with external resources</li> </ol>					
<b>11. Unit wise detailed content</b>					
<b>Unit-1</b>	<b>Number of lectures = 9</b>				
<ol style="list-style-type: none"> <li>1. Familiarization with concept of IOT, Arduino/Raspberry Pi and perform necessary software installation</li> <li>2. Study of different operating systems for Raspberry-Pi /Beagle board. Understanding the process of OS installation on Raspberry-Pi /Beagle board</li> <li>3. Study of Connectivity and configuration of Raspberry-Pi /Beagle board circuit with basic peripherals, LEDs. Understanding GPIO and its use in program.</li> <li>4. Use of Temperature Sensor and LED bar.</li> <li>5. Use of IR(Infra red) Sensor and LED bar</li> <li>6. Use of Camera.</li> <li>7. Use of X-Bee Module</li> <li>8. Use of Stepper Motor</li> <li>9. Client -Server Application</li> <li>10. Raspberry Pi to Cloud Interfacing.</li> </ol>					
<b>12. Brief Description of self-learning / E-learning component</b>					
The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University.					
The link to the E-Learning portal.					
<a href="https://elearning.sgtuniversity.ac.in/course-category/">https://elearning.sgtuniversity.ac.in/course-category/</a>					

## Architecting Smart IoT Devices

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	<b>Architecting smart IoT Devices</b>	<b>L</b>	<b>T</b>	<b>P</b>		
<b>3. Course Code</b>		<b>3</b>	<b>0</b>	<b>2</b>		
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE(✓)</b>	<b>OE ()</b>		
<b>5. Pre-requisite (if any)</b>		<b>6. Frequency (use tick marks)</b>	Even ( )	Odd (✓)	Either Sem()	Every Sem ()
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 0</b>		
<b>8. Course Description</b>						
<ul style="list-style-type: none"> <li>• This course will teach you how to develop an embedded systems device. In order to reduce the time to market, many pre-made hardware and software components are available today.</li> </ul>						
<b>17. Learning Objectives:</b>						
<ol style="list-style-type: none"> <li>1. Identify different IoT Applications with IoT architecture.</li> <li>2. Identify, test and interconnect components/parts of IoT system.</li> <li>3. . Identify and test various parts of embedded system.</li> <li>4. Identify and select various types of sensors used in Smart City.</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
<ol style="list-style-type: none"> <li>1. Identify and test Smart Lighting system and its components</li> <li>2. Identify, select, install and troubleshoot different module / devices used in SMART Street Light based on IoT and Cloud Technology.</li> <li>3. Identify, select, install and troubleshoot different module / devices used in SMART Parking</li> <li>4. Identify, select, install and troubleshoot different module / devices used in SMART Traffic.</li> </ol>						
<b>11. Unit wise detailed content</b>						
<b>Unit-1</b>	<b>Number of lectures = 9</b>					
<b>Fundamentals of Iot</b>						
Evolution of Internet of Things – Enabling Technologies – IoT Architectures: oneM2M, IoT World Forum (IoTWF) and Alternative IoT Models – Simplified IoT Architecture and Core IoT Functional Stack – Fog,						
<b>Unit – 2</b>	<b>Number of lectures = 9</b>					
Edge and Cloud in IoT – Functional Blocks of an IoT Ecosystem -Sensors, Actuators, and Smart Objects – Open Hardware Platforms for IoT.						
<b>Unit – 3</b>	<b>Number of lectures = 9</b>					
Routing over Low Power and Lossy Networks (RPL) – Application Transport Methods: Application Layer Not Present, Supervisory Control and Data Acquisition (SCADA) -Application Layer Protocols: CoAP and MQTT – Service discovery – mDNS.						



<b>Unit – 4</b>	<b>Number of lectures = 9</b>	
<p>Smart and Connected Cities: Street Layer, City Layer, Data Center Layer and Services Layer, Street Lighting, Smart Parking Architecture and Smart Traffic Control – Smart Transportation – Connected Cars.</p>		
<p><b>12. Brief Description of self-learning / E-learning component</b>  The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University.  The link to the E-Learning portal.  <a href="https://elearning.sgtuniversity.ac.in/course-category/">https://elearning.sgtuniversity.ac.in/course-category/</a></p>		
<p><b>13. Books Recommended</b></p>		
<p><b>Text Books</b></p> <ol style="list-style-type: none"> <li>2. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton, Jerome Henry, “IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things”, CISCO Press, 2017.</li> </ol>		
<p><b>14. Reference Books</b></p>		
<ol style="list-style-type: none"> <li>1. Perry Lea, “Internet of things for architects”, Packt, 2018.</li> <li>2. Jan Ho”ller, VlasiosTsiatsis, Catherine Mulligan, Stamatis, Karnouskos, Stefan Avesand, David Boyle, “From Machine-to-Machine to the Internet of Things -Introduction to a New Age of Intelligence”, Elsevier, 2014.</li> <li>3. Olivier Hersent, David Boswarthick, Omar Elloumi , “The Internet of Things – Key Applications and Protocols”, Wiley, 2012.</li> <li>4. Dieter Uckelmann, Mark Harrison, Michahelles, Florian (Eds), “Architecting the Internet of Things”, Springer, 2011.</li> </ol>		

## Architecting smart IoT Devices Lab

<b>1. Name of the Department:</b> Computer Science & Engineering						
<b>2. Course Name</b>	Architecting smart IoT Devices Lab	<b>L (3)</b>	<b>T (0)</b>		<b>P (2)</b>	
<b>3. Course Code</b>						
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE(✓)</b>		<b>OE ()</b>	
<b>5. Pre-requisite (if any)</b>		<b>6. Frequency (use tick marks)</b>	Even ()	Odd (✓)	Either Sem ()	Every Sem ()
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
Lectures = 0		Tutorials = 0	Practical = 48			
<b>8. Brief Syllabus</b>						
This course will teach you how to develop an embedded systems device. In order to reduce the time to market, many pre-made hardware and software components are available today.						
<b>18. Learning Objectives:</b>						
<ol style="list-style-type: none"> <li>1. Identify different IoT Applications with IoT architecture.</li> <li>2. Identify, test and interconnect components/parts of IoT system.</li> <li>3. Identify and test various parts of embedded system.</li> <li>4. Identify and select various types of sensors used in Smart City.</li> </ol>						
<b>10 Course Outcomes (COs):</b>						
<ol style="list-style-type: none"> <li>1. Identify and test Smart Lighting system and its components</li> <li>2. Identify, select, install and troubleshoot different module / devices used in SMART Street Light based on IoT and Cloud Technology.</li> <li>3. Identify, select, install and troubleshoot different module / devices used in SMART Parking</li> <li>4. Identify, select, install and troubleshoot different module / devices used in SMART Traffic.</li> </ol>						

### 11. Lab Experiment

Sr. No.	Title
1	Development Tools and Environments. Debugging Basics. Debugging Specials.
2	Real-Time Scheduling. Synchronisation and Communication web tour. Device Drivers. Multithreading Design.
3	Hardware & Software for EmS
4	Study of a few Embedded Processor Families. MCU, SOC, FPGA. Cache, pipeline and coupling
5	Networks. Software Components
6	OS for IoT Evaluation reports on the embedded OS

### 12. Brief Description of self-learning / E-learning component

## Application of IoT in Robotics

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	<b>Application of IoT in Robotics</b>	<b>L</b>	<b>T</b>	<b>P</b>		
<b>3. Course Code</b>		<b>3</b>	<b>0</b>	<b>2</b>		
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE(✓)</b>		<b>OE ()</b>	
<b>5. Pre-requisite (if any)</b>		<b>6. Frequency (use tick marks)</b>	Even (✓)	Odd ()	Either Sem()	Every Sem ()
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 0</b>		
<b>8. Course Description</b>						
This course will teach you how to develop an embedded systems device. In order to reduce the time to market, many pre-made hardware and software components are available today.						
<b>19. Learning Objectives:</b>						
<ol style="list-style-type: none"> <li>1. Identify different IoT Applications with IoT architecture.</li> <li>2. Identify, test and interconnect components/parts of IoT system.</li> <li>3. . Identify and test various parts of embedded system.</li> <li>4. Identify and select various types of sensors used in Smart City.</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
<ol style="list-style-type: none"> <li>1. Identify and test Smart Lighting system and its components</li> <li>2. Identify, select, install and troubleshoot different module / devices used in SMART Street Light based on IoT and Cloud Technology.</li> <li>3. Identify, select, install and troubleshoot different module / devices used in SMART Parking</li> <li>4. Identify, select, install and troubleshoot different module / devices used in SMART Traffic.</li> </ol>						
<b>11. Unit wise detailed content</b>						
<b>Unit-1</b>	<b>Number of lectures = 9</b>					
What is the IoT and why is it important? Elements of an IoT ecosystem, Technology drivers, Business drivers, Trends and implications, Overview of Governance, Privacy and Security Issues						
<b>Unit – 2</b>	<b>Number of lectures = 9</b>					
Protocol Standardization for IoT –Efforts –M2M and WSN Protocols –SCADA and RFIDProtocols –Issues with IoT Standardization –Unified Data Standards –Protocols – IEEE802.15.4–BACNet Protocol–Modbus –KNX –Zigbee–Network layer –APS layer – Security						
<b>Unit – 3</b>	<b>Number of lectures = 9</b>					
IoT Open source architecture (OIC)-OIC Architecture & Design principles-IoT Devices and deployment models-IoTivity : An Open source IoT stack -Overview-IoTivity stack architecture-Resource model and Abstraction.						
<b>3. Unit – 4</b>	<b>Number of lectures = 9</b>					

IoT applications for industry: Future Factory Concepts, Brownfield IoT, Smart Objects, Smart Applications. Study of existing IoT platforms /middleware, IoT-A, Hydra etc.

### **12. Brief Description of self-learning / E-learning component**

The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University.

The link to the E-Learning portal.

<https://elearning.sgtuniversity.ac.in/course-category/>

### **13. Books Recommended**

#### **Text Books**

4. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton, Jerome Henry, “IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things”, CISCO Press, 2017.

### **14. Reference Books**

5. Perry Lea, “Internet of things for architects”, Packt, 2018.
6. Jan Ho”ller, VlasiosTsiatsis, Catherine Mulligan, Stamatis, Karnouskos, Stefan Avesand, David Boyle, “From Machine-to-Machine to the Internet of Things -Introduction to a New Age of Intelligence”, Elsevier, 2014.
7. Olivier Hersent, David Boswarthick, Omar Elloumi , “The Internet of Things – Key Applications and Protocols”, Wiley, 2012.
8. Dieter Uckelmann, Mark Harrison, Michahelles, Florian (Eds), “Architecting the Internet of Things”, Springer, 2011.

## Application of IoT in Robotics Lab

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	<b>Application of IoT in Robotics Lab</b>	<b>L</b>	<b>T</b>	<b>P</b>		
<b>3. Course Code</b>		<b>3</b>	<b>0</b>	<b>2</b>		
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE(✓)</b>		<b>OE ( )</b>	
<b>5. Pre-requisite (if any)</b>	Computer Basics	<b>6. Frequency (use tick marks)</b>	Even (✓)	Odd ( )	Either Sem( )	Every Sem ( )
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 0</b>		
<b>8. Course Description</b>						
This course will teach students how to use the iot application in real time application. Students will design different projects using Arduino/Raspberry						
<b>20. Learning Objectives:</b>						
<ol style="list-style-type: none"> <li>1. To understand the basic functions of Arduino/Raspberry</li> <li>2. To understand the application of iot in healthcare system.</li> <li>3. To understand the programming functions of .Arduino</li> <li>4. To facilitate the development and application of problem-solving skills instudents.</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
<p>The students will be able to:-</p> <ol style="list-style-type: none"> <li>1. Understand the basic components of robots.</li> <li>2. Differentiate types of robots and robot grippers.</li> <li>3. Programme a robot to perform tasks in industrial applications.</li> </ol>						
<b>11. Unit wise detailed content</b>						
<b>Unit-1</b>	<b>Number of lectures = 9</b>					
<ol style="list-style-type: none"> <li>1. Familiarization with Arduino/Raspberry PI and perform necessary software installation</li> <li>2. To interface LED/Buzzer with Arduino/Raspberry Pi and write a program to turn ON LED for 1 sec after every 2 seconds</li> <li>3. To interface Push button/Digital sensor (IR/LDR) with Arduino/Raspberry Pi and write a program to turn ON LED when push button is pressed or at sensor detection.</li> <li>4. To interface DHT11 sensor with Arduina/Raspberry Pi and write a program to print temperature and humidity readings.</li> <li>5. To interface motor using relay with Arduino/Raspberry Pi and write a program to turn ON motor when push button is pressed</li> <li>6. Students are encouraged to explore the surrounding problems and design solutions based on IoT for any 3 of the following. <ol style="list-style-type: none"> <li>i. Health Monitoring System</li> <li>ii. Smart Irrigation System using Cloud</li> <li>iii. Smart Waste Management System</li> <li>iv. Smart Street Lights System</li> <li>v. Fire and Smoke Detection System</li> <li>vi. Gas Leakage Detection System vii. Water level Monitoring and Controlling System in Tanks</li> <li>viii. Air and Sound Pollution Monitoring System</li> <li>ix. Intruder (Human/Animal) Detection in Agriculture Field</li> <li>x. Girl Child Safety System using GPS and GSM</li> </ol> </li> </ol>						

## ROBOTICS LAB EXPERIMENTS.

1. Programming a simple Robot on Wheels.
2. Programming a Walking Robot.
3. Experiments based on Bipedal Robot.
4. Experiments based on Humanoid Robot
5. Spy Robot using RF Communication System
6. Study of robotic arm and its configuration
7. Study the robotic end effectors
8. Study of sensor integration.
9. Setting robot for any one industrial application after industrial visit.

# **Blockchain**

## Blockchain and Distributed Ledger Technology

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	Blockchain and Distributed Ledger Technology	<b>L</b>	<b>T</b>	<b>P</b>		
<b>3. Course Code</b>		<b>3</b>	<b>0</b>	<b>2</b>		
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE(✓)</b>		<b>OE ()</b>	
<b>5. Pre-requisite (if any)</b>		<b>6. Frequency (use tick marks)</b>	Even ()	Odd (✓)	Either Sem()	Every Sem ()
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 0</b>		
<b>8. Course Description</b>						
Blockchain technology and distributed ledgers have been hailed as a turning point in scaling information technology services at a global level. Although the digital currency Bitcoin is the best-known Blockchain application today, the technology is set to play a much broader role in cyber security innovation.						
<b>21. Learning Objectives:</b>						
<ol style="list-style-type: none"> <li>1. Understand what is a blockchain and a distributed ledger</li> <li>2. Develop or extend the ability to think critically about cybersecurity</li> <li>3. Understand the challenges of scaling information technology services across organizational barriers and at a global level.</li> <li>4. Analyse the security of basic cryptographic primitives like hash functions and digital signatures</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
<ol style="list-style-type: none"> <li>1. Understand how blockchain systems (mainly Bitcoin and Ethereum) work.</li> <li>2. To securely interact with them.</li> <li>3. Design, build, and deploy smart contracts and distributed applications.</li> <li>4. Integrate ideas from blockchain technology into their own projects</li> </ol>						
<b>11. Unit wise detailed content</b>						
<b>Unit-1</b>	<b>Number of lectures = 9</b>	Introduction to Blockchain Technology				
<b>Introduction to Blockchain</b>						
Blockchain concepts, evolution, structure, characteristics, a sample blockchain application, the blockchain stack, benefits and challenges, What is a Blockchain, Public Ledgers, Blocks in a Blockchain, Blockchains as public ledgers, Transactions, Distributed consensus. Building a block: Elements of Cryptography-Cryptographic Hash functions, Merkle Tree, Elements of Game Theory.						
<b>Unit – 2</b>	<b>Number of lectures = 9</b>	Satoshi’s Bitcoin				
<b>Blockchain Architecture and Use cases</b>						
Design methodology for blockchain applications, blockchain application templates, blockchain application development, Ethereum, Solidity, Sample use cases from Industries, Business problems.						
<b>Unit – 3</b>	<b>Number of lectures = 9</b>	<b>The Bitcoin Network</b> and Advanced Theories				
<b>Decentralized applications (Dapps)</b>						
Dapps, implementing Dapps, Ethereum Dapps, case studies related to Dapps, Byzantine fault tolerance, proof-of-work vs proof-of-stake, Security and Privacy of Blockchains, smart contract vulnerabilities, Scalability of Blockchains						



<b>Unit – 4</b>	<b>Number of lectures = 9</b>	Ethereum Clients
<b>Distributed Ledger Technology</b> Distributed Database, Two General Problem, Byzantine General problem and Fault Tolerance, Hadoop Distributed File System, Distributed Hash Table, ASIC resistance, Turing Complete. Cryptography: Hash function, Digital Signature - ECDSA, Memory Hard Algorithm, Zero Knowledge Proof.		
<b>12. Brief Description of self-learning / E-learning component</b> The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University.		
<b>13. Books Recommended</b>		
<b>Text Books</b> Blockchain applications: a hands-on approach, Bahga A., Madiseti V., VPT, 2017.		
<b>14. Reference Books</b>		
<ol style="list-style-type: none"> <li>1. Beginning Blockchain, A Beginner’s Guide to Building Blockchain Solutions, Bikramaditya Singhal, Gautam Dhameja, Priyansu Sekhar Panda, Apress, 2018.</li> <li>2. Blockchain A Practical Guide to Developing Business, Law, and Technology Solutions, Joseph J. Bambara and Paul R. Allen, McGraw Hill, 2018.</li> <li>3. Blockchain enabled Applications Vikram Dhillon, David Metcalf and Max Hooper, Apress, 2017,</li> <li>4. The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology, William Mougayar, Wiley, 2016.</li> <li>5. Blockchain Science: Distributed Ledger Technology, Roger Wattenhofer, Inverted Forest Publishing; 3rd edition, 2019.</li> </ol>		

## Blockchain and Distributed Ledger Technology Lab

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	Blockchain and Distributed Ledger Technology Lab	L	T	P		
<b>3. Course Code</b>		<b>3</b>	<b>0</b>	<b>2</b>		
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE(✓)</b>	<b>OE ()</b>		
<b>5. Pre-requisite (if any)</b>	Computer Basics	<b>6. Frequency (use tick marks)</b>	Even ()	Odd (✓)	Either Sem()	Every Sem ()
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 0</b>		
<b>8. Course Description</b>						
This course is designed to explore various aspects Distributed ledger technology like application in various domains.						
<b>22. Learning Objectives:</b>						
<ol style="list-style-type: none"> <li>1. Impart strong technical understanding of Blockchain technologies</li> <li>2. Develop familiarity of current technologies, tools, and implementation strategies</li> <li>3. Introduce application areas, current practices, and research activity</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
<p style="text-align: center;">The students will be able to:-</p> <ol style="list-style-type: none"> <li>1. Blockchain technology landscape</li> <li>2. Applications and implementation strategies</li> <li>3. State-of-the-art, open research challenges, and future direction</li> </ol>						
<b>11. List of Experiment</b>						
<ol style="list-style-type: none"> <li>1. Create a Simple Blockchain in any suitable programming language.</li> <li>2. Use Geth to Implement Private Ethereum Block Chain.</li> <li>3. Build Hyperledger Fabric Client Application.</li> <li>4. Build Hyperledger Fabric with Smart Contract.</li> <li>5. Create Case study of Block Chain being used in illegal activities in real world.</li> <li>6. Using Python Libraries to develop Block Chain Application.</li> <li>7. Write a program to generate Hash key.</li> <li>8. Using Java Libraries to develop Block Chain Applications.</li> <li>9. Write a program to create public key in Blockchain.</li> <li>10 Write a program to create private Key in Blockchain.</li> </ol>						

## Design and Development of Blockchain Applications

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	<b>Design and Development of Blockchain Applications</b>	<b>L</b>	<b>T</b>	<b>P</b>		
<b>3. Course Code</b>		<b>3</b>	<b>0</b>	<b>2</b>		
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE(✓)</b>		<b>OE ()</b>	
<b>5. Pre-requisite (if any)</b>		<b>6. Frequency (use tick marks)</b>	Even ( )	Odd (✓)	Either Sem()	Every Sem ()
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 0</b>		
<b>8. Course Description</b>						
This course provide the fundamental design and architectural primitives of Blockchain, the system and the security aspects, along with various use cases from different application domains.						
<b>23. Learning Objectives:</b>						
<ol style="list-style-type: none"> <li>1. Understand how blockchain systems (mainly Bitcoin and Ethereum) work,</li> <li>2. To securely interact with them,</li> <li>3. Design, build, and deploy smart contracts and distributed applications,</li> <li>4. Integrate ideas from blockchain technology into their own projects.</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
<p>The students will be able to:-</p> <ol style="list-style-type: none"> <li>1. Design, build, and deploy blockchain applications</li> <li>2. To understand the technology behind blockchain</li> <li>3. To comprehend the issues related to blockchain</li> <li>4. To study the real-world applications of blockchain</li> </ol>						
<b>11. Unit wise detailed content</b>						
<b>Unit-1</b>	<b>Number of lectures = 9</b>					
Blockchain Technology Blockchain Basics, Blockchain Evolution, Blockchain Structure, Blockchain Characteristics, Blockchain Application Example: Escrow, Blockchain Stack: Decentralized Computation Platform-Ethereum, Decentralized Storage Platform: Swarm, Decentralized Messaging Platform-Whisper, Smart Contracts, Decentralized Applications, Tools and Interfaces.						
<b>Unit – 2</b>	<b>Number of lectures = 9</b>					
Domain Specific Blockchain Applications Blockchain Applications: Internet of Things, Medical Record Management System, FinTech, Industrial and Manufacturing, Domain Name Service and future of Blockchain.						
<b>Unit – 3</b>	<b>Number of lectures = 9</b>					
Blockchains for real-world Applications Manufacturing and production, supply chain management, logistics and transportation, Internet of things, e-voting, healthcare, product life cycle, knowledge and innovation management, new business models and applications						

<b>Unit – 4</b>	<b>Number of lectures = 9</b>	
<p>Blockchain Components and Applications Templates  Blockchain Application Components, Design Methodology for Blockchain Applications, Blockchain Application Templates: Many to one, Many to one for IoT applications, Many to many or Peer to Peer, One to One for Financial Applications.</p>		
<p><b>12. Brief Description of self-learning / E-learning component</b>  The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University.  The link to the E-Learning portal.</p>		
<p><b>13. Books Recommended</b></p>		
<p><b>Text Books</b>  1. Blockchain Applications: A Hands-On Approach “ArshdeepBahga, Vijay Madiseti”.</p>		
<p><b>14. Reference Books</b></p>		
<p>1. Architecture for Blockchain Applications, Xu, Xiwei, Weber, Ingo, Staples, Mark.</p>		

## Design and Development of Blockchain Applications Lab

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	<b>Design and Development of Blockchain Applications Lab</b>	<b>L</b>	<b>T</b>	<b>P</b>		
<b>3. Course Code</b>		<b>3</b>	<b>0</b>	<b>2</b>		
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE(✓)</b>	<b>OE ()</b>		
<b>5. Pre-requisite (if any)</b>	Computer Basics	<b>6. Frequency (use tick marks)</b>	Even ()	Odd (✓)	Either Sem()	Every Sem ()
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 0</b>		
<b>8. Course Description</b>						
This course is designed to explore various aspects of Blockchain technology like application in various domains.						
<b>24. Learning Objectives:</b>						
<ol style="list-style-type: none"> <li>1. To provide conceptual understanding of blockchain technology.</li> <li>2. To familiarise with blockchain based solutions to innovate and improve business processes.</li> <li>3. To introduce design and architectural primitives of blockchain.</li> <li>4. To present system and security aspects of blockchain.</li> <li>5. To provide use cases from different application domains.</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
<p>The students will be able to:-</p> <ol style="list-style-type: none"> <li>1. Understand blockchain technology.</li> <li>2. Develop blockchain based solutions.</li> <li>3. Write smart contract using Hyperledger Fabric and Ethereum frameworks.</li> <li>4. Build and deploy blockchain applications for on-premise and cloud based architecture.</li> </ol>						
<b>11. Unit wise detailed content</b>						
<b>Unit-1</b>	<b>Number of lectures = 9</b>					
<ol style="list-style-type: none"> <li>1. Install and understand Docker container, Node.js, Java and Hyperledger Fabric, Ethereum and perform necessary software installation on local machine/create instance on cloud to run.</li> <li>2. Create and deploy a blockchain network using Hyperledger Fabric SDK for Java Set up and initialize the channel, install and instantiate chain code, and perform invoke and query on your blockchain network.</li> <li>3. Interact with a blockchain network. Execute transactions and requests against a blockchain network by creating an app to test the network and its rules.</li> <li>4. Deploy an asset-transfer app using blockchain. Learn app development within a Hyperledger Fabric network.</li> <li>5. Use blockchain to track fitness club rewards. Build a web app that uses Hyperledger Fabric to track and trace member rewards.</li> <li>6. Car auction network: A Hello World example with Hyperledger Fabric Node SDK and IBM Blockchain Starter Plan. Use Hyperledger Fabric to invoke chain code while storing results and data in the starter plan.</li> <li>7. Develop an IOT asset tracking app using Blockchain. Use an IOT asset tracking device to improve a supply chain by using Blockchain, IOT devices and Node-RED.</li> <li>8. Secure art using blockchain digital certificates. Node.js-based auction application can help democratize the art market.</li> </ol>						

## **12. Brief Description of self-learning / E-learning component**

The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University.

The link to the E-Learning portal.

## **13. Books Recommended**

### **Text Books**

- Melanie Swan, “Blockchain: Blueprint for a New Economy”, O’Reilly, 2015.
- Andreas Antonopoulos, “Mastering Bitcoin: Unlocking Digital Cryptocurrencies”, O’Reilly, 2014.

## **14. Reference Books**

1. Iran Bashir “Mastering Blockchain”, Second Edition Paperback 2018.
2. Daniel Drescher, “Blockchain Basics”, First Edition, Apress, 2017.
3. Ritesh Modi, “Solidity Programming Essentials: A Beginner’s Guide to Build Smart Contracts for Ethereum and Blockchain”, Packt Publishing.

### **Web Resources:**

<https://github.com/hyperledger/https://docs.docker.com/get-started/>  
<https://console.ng.bluemix.net/docs/services/blockchain/index.html> 2.  
<https://developer.ibm.com/patterns/create-and-deploy-blockchainnetwork/> 3.  
<https://developer.ibm.com/patterns/create-and-deploy-blockchainnetwork-using-fabric-sdk-java/> 4.  
[https://console.bluemix.net/docs/containers/container\\_index.html#container\\_index](https://console.bluemix.net/docs/containers/container_index.html#container_index) 5.  
<http://developer.ibm.com/patterns/fitness-club-rewards-points-iotand-retail-integration/> 6.  
<http://developer.ibm.com/patterns/car-auction-network-hyperledgerfabric-node-sdk-starter-plan/> 7.  
<http://developer.ibm.com/patterns/devlop-an-iot-asset-tracking-appusing-blockchain/> 8.  
<http://developer.ibm.com/patterns/securing-art-using-blockchaindigital-certificates/>

## Programming Fundamentals : Golang and Solidity

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	<b>Programming Fundamentals : Golang and Solidity</b>	<b>L</b>	<b>T</b>	<b>P</b>		
<b>3. Course Code</b>		<b>3</b>	<b>0</b>	<b>2</b>		
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE(✓)</b>	<b>OE ( )</b>		
<b>5. Pre-requisite (if any)</b>	Computer Basics	<b>6. Frequency (use tick marks)</b>	Even (✓)	Odd ( )	Either Sem( )	Every Sem ( )
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 0</b>		
<b>8. Course Description</b>						
Golang is next generation multi-purpose programming language that allows different users to create applications of various domains. Students will be able to learn primary fundamentals of Go and Solidity programming and potential of Go is to achieve modern computing requirements.						
<b>25. Learning Objectives:</b>						
9. Learning objectives						
<ol style="list-style-type: none"> <li>1. The objective of this course is to teach students the concepts of Statistics, probability,</li> <li>2. probability distribution, and other statistical methods to solve various engineering problem</li> <li>3. Master the fundamentals of writing Go and Solidity</li> <li>4. Learn core Golang and Solidity such as variables and flow control structures.</li> <li>5. Discover how to work with lists and sequence data.</li> <li>6. Write Go and Solidity functions to facilitate code reuse.</li> <li>7. V. Use Golang and Solidity to read and write files</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
The students will be able to:-						
<ol style="list-style-type: none"> <li>1. To acquire programming skills in core Golang and Solidity .</li> <li>2. To acquire Object Oriented Skills in Golang and Solidity.</li> <li>3. To develop the skill of designing Graphical user Interfaces in Golang and Solidity.</li> <li>4. To develop the ability to write database applications in Golang and Solidity.</li> </ol>						
<b>11. Unit wise detailed content</b>						
<b>Unit-1</b>	<b>Number of lectures = 9</b>					
Go – Overview, Environment Setup , Program Structure , Basic Syntax , Data Types , Variables , Constants , Operators , Decision Making , Loops , Functions , Scope Rules						
<b>Unit – 2</b>	<b>Number of lectures = 9</b>					
Go – Strings , Arrays , Pointers , Structures , Slice , Range , Maps , Recursion , Type Casting , Interfaces , Error Handling						
<b>Unit – 3</b>	<b>Number of lectures = 9</b>					
Solidity – Overview , Environment Setup , Basic Syntax , First Application , Comments , Types , Variables , Variable Scope , Operators , Loops , Decision Making , Strings , Arrays , Enums , Structs , Mappings , Conversions , Ether Units , Special Variables , Style Guide						

<b>Unit – 4</b>	<b>Number of lectures = 9</b>	
Solidity – Functions , Function Modifiers , View Functions , Pure Functions , Fallback Function, Function Overloading , Mathematical Functions , Withdrawal Pattern , Restricted Access , Contracts , Inheritance , Constructors , Abstract Contracts , Interfaces , Libraries , Assembly , Events , Error Handling		
<b>12. Brief Description of self-learning / E-learning component</b>		
The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University. The link to the E-Learning portal. <a href="https://elearning.sgtuniversity.ac.in/course-category/">https://elearning.sgtuniversity.ac.in/course-category/</a>		
<b>13. Books Recommended</b>		
<b>Text Books</b>		
I. Go Programming Language, The (Addison-Wesley Professional Computing Series)		
<b>14. Reference Books</b>		
I. An introduction to programming in Go : Caleb doxsey. II. Introducing Go: Build Reliable, Scalable Programs : Caleb Doxsey III. Solidity Programming Essentials: A beginner's guide to build smart contracts for Ethereum and blockchain		



## Programming Fundamentals : Golang and Solidity Lab

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	<b>Programming Fundamentals : Golang and Solidity Lab</b>	<b>L</b>	<b>T</b>	<b>P</b>		
<b>3. Course Code</b>		<b>3</b>	<b>0</b>	<b>2</b>		
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE(✓)</b>		<b>OE ()</b>	
<b>5. Pre-requisite (if any)</b>	Computer Basics	<b>6. Frequency (use tick marks)</b>	Even (✓)	Odd (□)	Either Sem()	Every Sem ()
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 0</b>		
<b>8. Course Description</b>						
<p>Student will learn about Solidity and Golang. It is a programming language for writing smart contracts which run on Ethereum Virtual Machine on Blockchain.</p>						
<b>26. Learning Objectives:</b>						
<ol style="list-style-type: none"> <li>1. To aware students about computer, its functions and utilities.</li> <li>2. To promote the development of computer-related skills for immediate application to other curricular areas.</li> <li>3. To provide a foundation for post-secondary education.</li> <li>4. To facilitate the development and application of problem-solving skills in students.</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
<p>The students will be able to:-</p> <ol style="list-style-type: none"> <li>1. Describe the usage of computers and why computers are essential components in business and society.</li> <li>2. Identify categories of programs, system software and applications. Organize and work with files and folders.</li> <li>3. Describe various types of networks network standards and communication software.</li> </ol>						
<b>11. Unit wise detailed content</b>						
<b>Unit-1</b>	<b>Number of lectures = 9</b>					
<ol style="list-style-type: none"> <li>1. Installing GO and First program of Hello World.</li> <li>2. Basic program for implementation of Data types and Golang Interface.</li> <li>3. Basic programs for variables, constants, ARRAY, FOR, IF ELSE , SWITCH.</li> <li>4. Basic programs for Golang Slice and append function and other functions.</li> <li>5. Basic programs on pointers, structures, packages , defer and stacking defers.</li> <li>6. Basic programs for methods, Concurrency, Goroutines and channels.</li> <li>7. Programs on SELECT, MUTEX , ERROR HANDLING</li> <li>8. Basic programs on Hello World, First App and implementation of Primitive Data Types</li> <li>9. Basic Programs on Variables, Reading and Writing to a State Variable , Ether and Wei , Gas and Gas Price.</li> <li>10. Programs on If / Else , For and While Loop , Mapping , Array , Enum , Structs.</li> <li>11. Basic programs on Data Locations - Storage, Memory and Call data , Function , View and Pure Functions , Error , Function Modifier , Events , Constructor.</li> <li>12. Basic program on Inheritance , Shadowing Inherited State Variables , Calling Parent Contracts , Visibility , Interface , Payable , Sending Ether - Transfer, Send, and Call</li> </ol>						

13. Basic programs on Fallback , Call , Delegatecall , Function Selector , Calling Other Contract.
14. Basic programs on Creating Contracts from a Contract , Try / Catch , , Import , Library , Hashing with Keccak256 , Verifying Signature
15. Applications : Multi Sig Wallet , Merkle Tree and Hacks : Re-Entrancy , Arithmetic Overflow and Underflow

**12. Brief Description of self-learning / E-learning component**

The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University.

The link to the E-Learning portal.

<https://elearning.sgtuniversity.ac.in/course-category/>

**13. Books Recommended**

**Text Books**

**14. Reference Books**

--

# **Data Analytics**

## Programming for Data Science

Name of the Department- Computer Science & Engineering						
<b>2. Course Name</b>	<b>Programming for Data Science</b>	L	T	P		
<b>3. Course Code (13470204)</b>		3	0	2		
<b>4. Type of Course (use tick mark)</b>		Core (✓)	PE(✓)		OE ( )	
<b>5. Pre-requisite (if any)</b>	NIL	<b>6. Frequency (use tick marks)</b>	Even ( )	Odd (✓)	Either Sem ( )	Every Sem ( )
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
Lectures = 36		Tutorials = 0	Practical = 0			
<b>8. Course Description</b>						
To provide necessary knowledge on how to manipulate data objects, produce graphics, analyse data using common statistical methods and generate reproducible statistical reports with programming in Python and R						
<b>9. Learning objectives:</b>						
<ol style="list-style-type: none"> <li>1. Apply principles of Data Science to the analysis of business problems.</li> <li>2. Use data mining software to solve real-world problems.</li> <li>3. Employ cutting edge tools and technologies to analyze Big Data.</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
<ol style="list-style-type: none"> <li>1. Ability to solve the analytical problems using Python and R</li> <li>2. Develop competency in the Python programming language and a number of data related Python libraries such as Pandas, Numpy, and Scipy</li> <li>3. Ability to communicate results of analysis effectively using visualizations in Python and R</li> <li>4. Import, export and manipulate data and produce statistical summaries of continuous and categorical data in Python and R</li> <li>5. Ability to perform exploratory data analysis using Python and R 17SLO</li> </ol>						
<b>11. Unit wise detailed content</b>						
<b>Unit-1</b>	<b>Number of lectures = 10</b>					
Expressions, Operators, matrices, Decision Statements in python, Control Flow and Functions in python, Classes, Objects, Packages and Files in python, Tuple, Lists, Sequences, Dictionaries, Comprehensions.						
<b>Unit – 2</b>	<b>Number of lectures = 9</b>					
Numpy Arrays objects, Creating Arrays, basic operations, Indexing, Slicing and iterating, copying arrays, shape manipulation, Identity array, eye function, Universal function, Linear algebra with Numpy, eigen values and eigen vectors with Numpy						
<b>Unit – 3</b>	<b>Number of lectures = 8</b>					
Aggregation and Joining, Pandas Object: Concatenating and appending data frames, index objects, Handling Time series data using pandas Handling missing values using pandas, Reading and writing the data including JSON data, Web scraping using python, Combining and merging, Datasets Data transformations Basic matplotlib plots, common plots used in statistical analysis in python.						

Unit – 4	Number of lectures = 9	
<p>Common plots used in statistical analysis in python Datatypes in R2. Sequence generation, Vector and subscript, Random2 number generation in R Data frames and R functions2 Data manipulation and Data Reshaping using plyr, dplyr,2 reshape2 Parametric statistics and Non-parametric statistics2 Continuous and Discrete Probability distribution using R2 Correlation and covariance, contingency tables2 Overview of Sampling, different sampling techniques2 R and data base connectivity2</p> <p>Web application development with R using Shiny2 Approaches to dealing with missing data in R2 Exploratory data analysis with simple visualizations using R 2 Feature or Attribute selection using R2 Dimensionality Reduction with R2 Time series data analysis with R2</p>		
<p>12. Brief Description of self-learning / E-learning component</p> <p>The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University.</p> <p>The link to the E-Learning portal.</p>		
<p>13. Books Recommended</p>		
<p>Text Books</p> <ol style="list-style-type: none"> <li>1. James Payne, “Beginning Python: Using Python 2.6 and Python 3.1” Wrox, Ist Edition, 2010</li> <li>2. Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, “Data Structures and Algorithms in Python”, John Wiley &amp; sons, 2013.</li> </ol>		
<p>Reference Books</p>		
<ol style="list-style-type: none"> <li>3. Ivan Idris, “Python Data Analysis”, Packt Publishing Limited, 2014</li> <li>4. Wes McKinney, “Python for Data Analysis Data Wrangling with Pandas, NumPy, and IPython”, O'Reilly Media, Ist Edition, 2012</li> <li>5. Michael Heydt, “Learning Pandas - Python Data Discovery and Analysis Made Easy”, Packt Publishing Limited, 2015.</li> <li>6. Jacqueline Kazil,Katharine Jarmul, “Data Wrangling with Python: Tips and Tools to MakeYour Life Easier”, O'Reilly Media, Ist Edition, 2016.</li> <li>7. <a href="https://docs.scipy.org/doc/numpy-dev/reference/index.html#reference">https://docs.scipy.org/doc/numpy-dev/reference/index.html#reference</a></li> <li>8. <a href="http://www.python-course.eu/numpy.php">http://www.python-course.eu/numpy.php</a></li> <li>9. Michael J. Crawley, “The R Book”, Wiley, 2nd Edition, 2012.</li> <li>10. Robert Kabacoff, “R in Action”, Manning Publication, Ist Edition, 2011.</li> <li>11. TorstenHothorn, Brian S. Everitt, “A Handbook of Statistical Analyses Using R”, Chapman and Hall_CRC, 2nd Edition, 2009.</li> <li>12. Chris Beeley "Web Application Development with R Using Shiny", Pact Publishing, 2013.</li> <li>13. Phil Spector, “Data Manipulation with R”, Springer, 2008.</li> <li>14. Prabhanjan N. Tattar, Suresh Ramaiah, B. G. Manjunath, “ A Course in Statistics with R”, wiley, 2016</li> <li>15. PawelCichosz, “Data Mining Algorithms: Explained Using R”, wiley, 2014</li> </ol>		

## Programming for Data Science Lab

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	<b>Programming for Data Science Lab</b>	<b>L</b>	<b>T</b>		<b>P</b>	
<b>3. Course Code</b>		<b>3</b>	<b>0</b>		<b>2</b>	
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE(✓)</b>		<b>OE ()</b>	
<b>5. Pre-requisite (if any)</b>	Programming	<b>6. Frequency (use tick marks)</b>	Even ()	Odd (✓)	Either Sem()	Every Sem ()
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 0</b>		<b>Tutorials = 0</b>		<b>Practical = 10</b>		
<b>8. Course Description</b>						
This course introduces you to the basics of the R language such as data types, techniques for manipulation, and how to implement fundamental programming tasks.						
<b>27. Learning Objectives:</b>						
<ol style="list-style-type: none"> <li>1. Manipulate numeric and textual data types using the R programming language</li> <li>2. Manipulate numeric and textual data types using the R programming language</li> <li>3. Define and manipulate R data structures, including vectors, factors, lists, and data frames</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
The students will be able to:-						
<ol style="list-style-type: none"> <li>1. Understand the basics in R programming in terms of constructs, control statements, string functions</li> <li>2. Understand the use of R for Big Data analytics</li> <li>3. Learn to apply R programming for Text processing</li> <li>4. Able to appreciate and apply the R programming from a statistical perspective</li> </ol>						
<b>11. Lab Experiments</b>						
Sr.No	Topics					
1	Creating and displaying Data.					
2	Matrix manipulations					
3	Creating and manipulating a List and an Array					
4	Creating a Data Frame and Matrix-like Operations on a Data Frame					
5	Merging two Data Frames					
6	Applying functions to Data Frames					
7	Using Functions with Factors					
8	Accessing the Internet					
9	String Manipulations					
10	Visualization Effects					
11	Plotting with Layers					Overriding
12	Overriding Aesthetics					
13	Histograms and Density Charts					
14	Simple Linear Regression – Fitting, Evaluation and Visualization					
15	Multiple Linear Regression, Lasso and Ridge Regression					

## Big Data Technologies

Name of the Department- Computer Science & Engineering						
<b>2. Course Name</b>	<b>Big Data Technologies</b>	<b>L</b>	<b>T</b>		<b>P</b>	
<b>3. Course Code</b>		<b>3</b>	<b>0</b>		<b>2</b>	
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE(✓)</b>		<b>OE ()</b>	
<b>5. Pre-requisite (if any)</b>	<b>NIL</b>	<b>6. Frequency (use tick marks)</b>	<b>Even ( )</b>	<b>Odd (✓)</b>	<b>Either Sem ()</b>	<b>Every Sem ()</b>
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		<b>Tutorials = 0</b>		<b>Practical = 0</b>		
<b>8. Course Description</b>						
Provides an overview of machine learning techniques to explore, analyze, and leverage data. You will be introduced to tools and algorithms you can use to create machine learning models that learn from data, and to scale those models up to big data problems.						
<b>9. Learning objectives:</b>						
To have knowledge on accessing, storing and manipulating the huge data from different resources. Working environment of Pig and Hive for processing the structured and unstructured data and differentiate the RDBMS and Hive architectures and implement queries to process the data using sqoop. 4. To have a knowledge on searching mechanisms using solr.						
<b>10. Course Outcomes (COs):</b>						
1. Illustrate the usage of data on different Big data ecosystems. 2. Demonstrate the Pig architecture and evaluation of pig scripts. 3. Describe the Hive architecture and execute SQL queries on sample data sets. 4. Understand the process of transferring data between different file systems and to execute operations using sqoop. 5. Understand the concepts of indexing and use these concepts in solr search engine. 6. Implement and evaluate the data manipulation procedures using pig, hive, sqoop and solr. 7. Develop an application using different eco system tools by taking standard sample data set.						
<b>11. Unit wise detailed content</b>						
<b>Unit-1</b>	<b>Number of lectures = 10</b>					
Big data- Concepts, Needs and Challenges of big data. Types and source of big data. Components of Hadoop Eco System- Data Access and storage, Data Intelligence, Data Integration, Data Serialization, Monitoring, Indexing. Introduction, Parallel processing using Pig, Pig Architecture, Grunt, Pig Data Model-scalar and complex types. Pig Latin- Input and output, Relational operators, User defined functions. Working with scripts.						
<b>Unit – 2</b>	<b>Number of lectures = 9</b>					
Introduction-Hive modules, Data types and file formats, Hive QL-Data Definition and Data Manipulation. Hive QL queries, Hive QL views- reduce query complexity. Hive scripts. Hive QL Indexes-create, show, drop. Aggregate functions. Bucketing vs Partitioning						
<b>Unit – 3</b>	<b>Number of lectures = 8</b>					
Relational database management in Hadoop: Bi directional data transfer between Hadoop and external database. Import data- Transfer an entire table, import subset data, use different file format. Incremental import import new data, incrementally import data, preserving the value. Export transfer data from Hadoop, update the data, update at the same time, export subset of						

columns. Hadoop ecosystem integration- import data to hive, using partitioned hive tables, replace special delimiters.		
<b>Unit – 4</b>	<b>Number of lectures = 9</b>	
Introduction. Information retrieval search engine, categories of data, inverted index. Design-field attributes and types. Indexing- indexing tool. Indexing operations using csv documents. Searching data- parameters, default query, Recent Trends in Big data		
12. Brief Description of self-learning / E-learning component The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University. The link to the E-Learning portal.		
13. Books Recommended		
<b>Text Books</b>		
1. Alan Gates, Programming Pig Dataflow Scripting with Hadoop, O'Reilly Media, Inc, 2011.		
2. Jason Rutherglen, Dean Wampler, Edward Capriolo, Programming Hive, O'Reilly Media Inc, 2012		
3. Kathleen Ting, Jarek Jarcec Cecho, Apache Sqoop Cookbook, O'Reilly Media Inc, 2013.		
<b>Reference Books</b>		
4. Dikshant Shahi, Apache Solr: A Practical approach to enterprise search, Apress, 2015.		
5. Chuck Lam, Hadoop in Action, Manning Publications, 2010.		
6. Andrea Gazzarini, Apache Solr Essentials, PACKT Publications, 2015.		



## Big Data Technologies Lab

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	<b>Big Data Technologies Lab</b>	<b>L</b>	<b>T</b>		<b>P</b>	
<b>3. Course Code</b>		<b>3</b>	<b>0</b>		<b>2</b>	
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE(✓)</b>		<b>OE ( )</b>	
<b>5. Pre-requisite (if any)</b>		<b>6. Frequency (use tick marks)</b>	Even ( )	Odd (✓)	Either Sem ( )	Every Sem ( )
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 0</b>		<b>Tutorials = 0</b>		<b>Practical = 10</b>		
<b>8. Course Description</b>						
<p>This course will introduce the students to practical aspects of analytics at large scale, i.e., big data. The course will start with a basic introduction to big data concepts spanning hardware, systems and software, and then delve into the following topics.</p>						
<b>28. Learning Objectives:</b>						
<ol style="list-style-type: none"> <li>1. To understand setting up of Hadoop Cluster</li> <li>2. To solve problems using Map Reduce Technique</li> <li>3. To solve Big Data problems</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
<p>The students will be able to:-</p> <ol style="list-style-type: none"> <li>1. Set up multi-node Hadoop Clusters</li> <li>2. Apply Map Reduce algorithms for various algorithms</li> <li>3. Design new algorithms that uses Map Reduce to apply on Unstructured and structured data</li> </ol>						
<b>11. Lab Experiments</b>						
Sr.No	Topics					CO
1	Set up a pseudo-distributed, single-node Hadoop cluster backed by the Hadoop Distributed File System, running on Ubuntu Linux. After successful installation on one node, configuration of a multi-node Hadoop cluster(one master and multiple slaves).					1
2	MapReduce application for word counting on Hadoop cluster					
3	Unstructured data into NoSQL data and do all operations such as NoSQL query with API.					2
4	K-means clustering using map reduce					
5	Page Rank Computation					
6	Mahout machine learning library to facilitate the knowledge build up in big data analysis.					2,3
7	Application of Recommendation Systems using Hadoop/mahout libraries					

### Mining Massive Data

<b>Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	<b>Mining Massive Data</b>	L	T	P		
<b>3. Course Code</b>		3	0	2		
<b>4. Type of Course (use tick mark)</b>		Core (✓)	PE(✓)		OE ( )	
<b>5. Pre-requisite (if any)</b>	<b>NIL</b>	6. Frequency (use tick marks)	Even (✓)	Odd ( )	Either Sem ( )	Every Sem ( )
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 36</b>		Tutorials = 0		Practical = 0		
<b>8. Course Description</b>						
Provides an overview of data mining and machine learning algorithms for analyzing very large amounts of data. The emphasis is on Map Reduce as a tool for creating parallel algorithms that can process very large amounts of data.						
9. Learning objectives:						
<ol style="list-style-type: none"> <li>1. To provide comprehensive knowledge on developing and applying machine learning algorithms for massive real-world datasets in distributed frameworks.</li> <li>2. To demonstrate the use of big data analytics tools like Spark and Mahout for mining massive datasets.</li> <li>3. To impart in depth knowledge on Deep Learning and Extreme Learning concepts.</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
<ol style="list-style-type: none"> <li>1. Identify right machine learning / mining algorithm for handling massive data</li> <li>1. 2. Apply classification and regression models with Spark and Mahout</li> <li>2. Implement clustering models using Spark and Mahout</li> <li>3. 4. Mine social Network graphs using MapReduce</li> <li>4. 5. Apply semi supervised learning for clustering and classification</li> <li>5. 6. Use deep learning to solve real-life problem</li> <li>6. Use Extreme Learning Machine for classification and regression.</li> <li>7. 8. Use big data analytics tools such as Spark, Mahout and H2O in solving problems based on Machine learning</li> </ol>						
11. Unit wise detailed content						
<b>Unit-1</b>	<b>Number of lectures = 10</b>					
MapReduce Based Machine Learning: Module content: K-Means, PLANET, Parallel SVM, Association Rule Mining in MapReduce, Inverted Index, Page Ranking, Expectation Maximization, Bayesian Networks Classification and Regression models with Spark and Mahout: Module content: Linear support vector machines - Naive Bayes model- Decision Trees - Least square regression- Decision trees for regression.						
<b>Unit – 2</b>	<b>Number of lectures = 9</b>					
Clustering in Spark and Mahout: Module content: Hierarchical Clustering in a Euclidean and Non-Euclidean Space - The Algorithm of Bradley, Fayyad, and Reina - A variant of K-means algorithm - Processing Data in BFR Algorithm CURE algorithm - Clustering models with Spark - Spectral clustering using Mahout						
<b>Unit – 3</b>	<b>Number of lectures = 8</b>					
Mining Social-Network Graphs: Module content: Clustering of Social-Network Graphs - Direct Discovery of Communities - Partitioning of Graphs Finding Overlapping Communities - Counting Triangles using MapReduce Neighborhood						

Properties of Graphs Semi-Supervised Learning: Module content: Introduction to Semi-Supervised Learning, Semi-Supervised Clustering, Transductive Support Vector Machines		
<b>Unit – 4</b>	<b>Number of lectures = 9</b>	
Deep Learning: Module content: Introduction, Deep Neural Networks, Deep Belief Networks, Auto Encoders, Recurrent Networks Extreme Learning: Extreme Learning Machines (ELM), ELM auto encoder, Extreme Support Vector Regression		
12. Brief Description of self-learning / E-learning component The students will be encouraged to learn using the SGT E-Learning portal and choose the relevant lectures delivered by subject experts of SGT University. The link to the E-Learning portal.		
13. Books Recommended		
Text Books 1. Jure Leskovec, AnandRajaraman, Je_ Ullman, "Mining of Massive Datasets", Stanford Press,2011. 2. Nick Pentreath, "Machine Learning with Spark", Packt Publishing, 3. Olivier Chapelle, Bernhard Scholkopf, Alexander Zien "Semi-Supervised Learning", The MIT Press,2006.		
<b>Reference Books</b>		
1. Ron Bekkerman, Mikhail Bilenko, John Langford "Scaling Up Machine Learning: Parallel and Distributed Approaches", Cambridge University Press, 2012. 2. Jimmy Lin, Chris Dyer, "Data-Intensive Text Processing with MapReduce", Morgan Claypool Publishers, 2010. 3. Hennessy, J.L. and Patterson, D.A., 2011. Computer architecture: a quantitative approach. Elsevier. 4. ChandramaniTiwary "Learning Apache Mahout", Packt Publishing, 2015. 5. Fuchen Sun, Kar-Ann Toh, Manuel Grana Romay, KezhiMao,"Extreme Learning Machines2013: Algorithms and Applications", Springer, 2014.		

## Mining Massive Data Lab

<b>1. Name of the Department- Computer Science &amp; Engineering</b>						
<b>2. Course Name</b>	<b>Mining Massive Data Lab</b>	<b>L</b>	<b>T</b>		<b>P</b>	
<b>3. Course Code</b>		<b>3</b>	<b>0</b>		<b>2</b>	
<b>4. Type of Course (use tick mark)</b>		<b>Core (✓)</b>	<b>PE(✓)</b>		<b>OE ( )</b>	
<b>5. Pre-requisite (if any)</b>		<b>6. Frequency (use tick marks)</b>	Even (✓)	Odd ( )	Either Sem( )	Every Sem ( )
<b>7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</b>						
<b>Lectures = 0</b>		<b>Tutorials = 0</b>		<b>Practical = 10</b>		
<b>8. Course Description</b>						
This course will introduce the students to Data mining. This refers to extracting or mining knowledge from large amounts of data. Data mining can also be referred as knowledge mining from data, knowledge extraction, data archeology and data dredging.						
<b>29. Learning Objectives:</b>						
<ol style="list-style-type: none"> <li>1. To familiarize students with the basic concepts of Data mining and Warehousing</li> <li>2. To explain and demonstrate various mining algorithms on real world data.</li> <li>3. To brief students about the future trends in the fields of data mining.</li> </ol>						
<b>10. Course Outcomes (COs):</b>						
The students will be able to:-						
<ol style="list-style-type: none"> <li>1. Describe various preprocessing techniques and statistical techniques and apply those techniques on the given data set.</li> <li>2. Apply various association rule mining algorithms on the given data set</li> <li>3. Apply various classification algorithms on the given data set.</li> </ol>						
<b>11. Lab Experiments</b>						
Sr.No	Topics					CO
1	Demonstration of preprocessing on .arff file using student data .arff					
2	To perform the statistical analysis of data					
3	Demonstration of association rule mining using apriory algorithm on supermarket data.					2
4	Demonstration of FP Growth algorithm on supermarket data					
5	To perform the classification by decision tree induction using weka tools.					
6	To perform classification using Bayesian classification algorithm using R.					
7	To perform the cluster analysis by k-means method using R.					
8	To perform the hierarchical clustering using R programming.					
9	Study of Regression Analysis using R programming.					
10	10. Outlier detection using R programming					