Scheme of Examination

and Syllabus

for

Master of Computer Applications

Batch 2019 Onwards

SGT University,

Gurgaon, Haryana

MCA (2019-2020)

Sem										L	T	P	hr/ week	Credits
I	Computer Organization and Architecture 3-1-0(4)	Problem Solving and Programming with C 3-0-0(3)	Computer and Internet Fundamentals 3-0-0(3)	Discrete Structure 3-1-0(4)	Human Values & Ethics 3-0-0(3)	Problem Solving and Programming with C Lab 0-0-4(2)	Web Development Lab using PHP 0-0-4(2)	Personality Development and Communication Skills – I 0-0-2(1)		15	2	10	27	22
II	Operating System 3-0-0(3)	Relational Data Base Management Systems 3-0-0(3)	Object Oriented Programming with C++ 3-0-0(3)	Data and File Structures 3-0-0(3)	PE-I 3-0-0(3)	Relational Data Base Management Systems Lab 0-0-2(1)	Object Oriented Programming with C++ Lab 0-0-4(2)	Data and File Structures Lab 0-0-4(2)	Operating System Lab 0-0-2(1)	15	0	12	27	21
III	Software Engineering 3-1-0(4)	Theory of Computation 3-1-0(4)	Data Communication and Networks 3-0-0(3)	Java Programming 3-1-0(4)	PE-II 3-0-0(3)		Data Communication and Networks Lab 0-0-4(2)	Java Programming Lab 0-0-4(2)	Colloquium I 0-0-2(1)	15	3	10	28	23
IV	Cloud Computing 3-0-0(3)	Computer Graphics and Multimedia 3-1-0(4)	Analysis & Design of Algorithms 3-1-0(4)	PE-III 3-0-0(3)	Artificial Intelligence 3-0-0(3)	Computer Graphics and Multimedia Lab 0-0-4(2)	Artificial Intelligence Lab 0-0-4(2)	Cloud Computing 0-0-2(1)	Colloquium II 0-0-2(1)	15	2	12	29	23
V	Advanced Web Technologies 3-0-0(3)	Software Project Management 3-0-0(3)	PE-IV 3-0-0(3)	PE-V 3-0-0(3)	PE-VI 3-0-0(3)	Software Testing 2-0-0(2)	Software Testing Lab 0-0-2(1)	Advanced Web Technologies Lab 0-0-2(1)	Minor Project 0-0-6(3)	17	0	10	27	22
VI	Dissertation 0-0-0(10)	Dissertation Seminar 0-0-0(5)								0	0	0	0	15
								Total		**77**	**7**	**54**	**138**	**126**

Subject Code	Program Electives	
13050205	Information Retrieval System	PE-I
13050206	Enterprise Resource Planning	PE-I
13050207	System Networking & Administration	PE-I
13050208	Organizational Behaviour	PE-I
13050209	Management Information System	PE-I
13050305	Enterprise Computing in Java	PE-II
13050306	Digital Signal Processing	PE-II
13050307	Advanced Database Management Systems	PE-II
13050308	Mobile Computing	PE-II
13050309	Advanced Computer Networks	PE-II
13050404	Distributed DBMS and Object Oriented Databases	PE-III
13050405	Microprocessor and Interfacing	PE-III
13050406	Operation Research	PE-III
13050407	Object Oriented Analysis and Design	PE-III
13050408	Software Quality Assurance	PE-III
13050504	Programming in Python	PE-IV
13050505	Distributed Operating Systems	PE-IV
13050506	Advanced Computer Architecture and Parallel Processing	PE-IV
13050507	Compiler Design	PE-IV
13050508	Embedded System	PE-IV
13050510	Computer Oriented Statistical Numerical Methods	PE-V
13050511	Business Process Management and Intelligence	PE-V
13050512	Data Warehousing and Data Mining	PE-V
13050513	IT Network Security	PE-V
13050515	IT System Security	PE-VI
13050516	Information Security Fundamentals	PE-VI
13050517	Ethical Hacking	PE-VI
13050518	Cyber Security	PE-VI

SGT University, Gurgaon, Haryana

Scheme of Master of Computer Applications for Batch 2019 onwards

Master of Computer Applications Semester – I

Paper Code	Title	L	T	P	Total	Int.	Ext.	Total	Credits
13050101	Computer Organization and Architecture	3	1	-	4	40	60	100	4
13050102	Problem Solving and Programming with C	3	-	-	3	40	60	100	3
13050103	Computer and Internet Fundamentals	3	-	-	3	40	60	100	3
13050104	Discrete Structure	3	1	-	4	40	60	100	4
13050105	Human Values & Ethics	3	-	-	3	40	60	100	3
PRACTICAL									
13050106	Problem Solving and Programming with C Lab	-	-	4	4	40	60	100	2
13050107	Web Development Lab using PHP	-	-	4	4	40	60	100	2
13050108	Personality Development and Communication Skills - I	-	-	2	2	40	60	100	1
	Total	**15**	**2**	**10**	**27**	**320**	**480**	**800**	**22**

Master of Computer Applications Semester – II

Paper Code	Title	L	T	P	Total	Int.	Ext.	Total	Credits
13050201	Operating System	3	-	-	3	40	60	100	3
13050202	Relational Data Base Management Systems	3	-	-	3	40	60	100	3
13050203	Object Oriented Programming with C++	3	-	-	3	40	60	100	3
13050204	Data and File Structures	3	-	-	3	40	60	100	3
	Program Electives I	3	-	-	3	40	60	100	3
PRACTICAL									
13050210	Relational Data Base Management Systems Lab	-	-	2	2	40	60	100	1
13050211	Data & File Structure-Lab	-	-	4	4	40	60	100	2
13050212	Object Oriented Programming with C++ Lab	-	-	4	4	40	60	100	2
13050213	Operating System Lab	-	-	2	2	40	60	100	1
	Total	**15**	-	**12**	**27**	360	540	900	**21**

Int: Internal, **Ext:** External

SGT University, Gurgaon, Haryana

Scheme of Master of Computer Applications for Batch 2019 onwards

Master of Computer Applications Semester – III

Paper Code	Title	L	T	P	Total	Int.	Ext.	Total	Credits
13050301	Software Engineering	3	1	-	4	40	60	100	4
13050302	Theory of Computation	3	1	-	4	40	60	100	4
13050303	Data Communication and Networks	3	-	-	3	40	60	100	3
13050304	Java Programming	3	1	-	4	40	60	100	4
	Program Electives II	3	-	-	3	40	60	100	3
PRACTICAL									
13050310	Data Communication & Networks Lab	-	-	4	4	40	60	100	2
13050311	Java Programming Lab	-	-	4	4	40	60	100	2
13050312	Colloquium I	-	-	2	2	40	60	100	1
	Total	**15**	**3**	**10**	**28**	**360**	**540**	**900**	**23**

Master of Computer Applications Semester – IV

Paper Code	Title	L	T	P	Total	Int.	Ext.	Total	Credits
13050401	Cloud Computing	3	-	-	3	40	60	100	3
13050402	Computer Graphics and Multimedia	3	1	-	4	40	60	100	4
13050403	Analysis & Design of Algorithms	3	1	-	4	40	60	100	4
	Program Electives III	3	-	-	3	40	60	100	3
13050409	Artificial Intelligence	3	-	-	3	40	60	100	3
PRACTICAL									
13050410	Computer Graphics & Multimedia Lab	-	-	4	4	40	60	100	2
13050411	Artificial Intelligence Lab	-	-	4	4	40	60	100	2
13050412	Cloud Computing Lab	-	-	2	2	40	60	100	1
13050413	Colloquium II	-	-	2	2	40	60	100	1
	Total	**15**	**2**	**12**	**28**	320	480	**800**	**23**

Int: Internal, **Ext:** External

SGT University, Gurgaon, Haryana

Scheme of Master of Computer Applications for Batch 2019 onwards

Master of Computer Applications Semester – V

Paper Code	Title	L	T	P	Total	Int.	Ext.	Total	Credits
13050501	Advanced Web Technologies	3	-	-	3	40	60	100	3
13050502	Software Project Management	3	-	-	3	40	60	100	3
13050503	Software Testing	2	-	-	2	40	60	100	2
	Program Electives IV	3	-	-	3	40	60	100	3
	Program Electives V	3	-	-	3	40	60	100	3
	Program Electives VI	3	-	-	3	40	60	100	3
13050519	Software Testing Lab	-	-	2	2	40	60	100	1
13050520	Advanced Web Tech. Lab	-	-	2	2	40	60	100	1
13050521	Minor Project	-	-	6	6	40	60	100	3
	Total	**18**	**-**	**10**	**27**	320	480	**800**	**22**

Int: Internal, **Ext:** External

SGT University, Gurgaon, Haryana

Scheme of Master of Computer Applications for Batch 2019 onwards

Master of Computer Applications Semester – VI

Paper Code	Title	L	T	P	Total	Int.	Ext.	Total	Credits
13050601	Dissertation	-	-	-	-	100	100	200	10
13050602	Dissertation Seminar	-	-	-	-	40	60	100	5
	Total	-	-	-	-	**140**	**160**	**300**	**15**

Int: Internal, **Ext:** External

Note: The student will submit a synopsis for approval from the departmental committee in a specified format. The student will have to present the progress of the work through seminars and progress reports.

SEMESTER III
Software Engineering

1.Name of the Department- Computer Science Engineering						
2.Course Name	**Software Engineering**	**L**	**T**		**P**	
3.Course Code		**3**	**0**		**4**	
4.Type of Course (use tick mark)		**Core (✓)**	**PE()**		**OE()**	
5.Pre-requisite (if any)		**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 = 48**			

8.Course Description

This course covers the fundamentals of software engineering, including understanding system requirements, finding appropriate engineering compromises, effective methods of design, coding, and testing, team software development, and the application of engineering tools.

9.Learning objectives

The program will prepare our students to be successful professionals in the field with solid fundamental knowledge of software engineering.

- **I.** Be successful professionals in the field with solid fundamental knowledge of software engineering
- **II.** Utilize and exhibit strong communication and interpersonal skills, as well as professional and ethical principles when functioning as members and leaders of multi-disciplinary teams
- **III.** Apply their foundations in software engineering to adapt to readily changing environments using the appropriate theory, principles and processes

10.Course Outcomes (COs):

I.	An ability to apply knowledge of mathematics, science, and engineering.
II.	An ability to design and conduct experiments, as well as to analyze and interpret data.
III.	An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.
IV.	An ability to function on multi-disciplinary teams.
V.	An ability to identify, formulates, and solves engineering problems.
VI.	An understanding of professional and ethical responsibility.
VII.	An ability to communicate effectively.

11.Unit wise detailed content

Unit-1	**Number of lectures = 08**	

Introduction: The process, software products, emergence of software engineering, evolving role of software, software life cycle models, Software Characteristics, Applications, Software crisis. Software project management: Project management concepts, software process and project metrics Project planning, project size estimation metrics, project estimation Techniques, empirical estimation

techniques, COCOMO- A Heuristic estimation techniques, staffing level estimation, team structures, staffing, risk analysis and management, project scheduling and tracking.

Unit – 2	**Number of lectures = 10**	

Requirements Analysis and specification requirements engineering, system modeling and simulation Analysis principles modeling, partitioning Software, prototyping: , Prototyping methods and tools; Specification principles, Representation, the software requirements specification and reviews Analysis Modeling: Data Modeling, Functional modeling and information flow: Data flow diagrams, Behavioral Modeling; The mechanics of structured analysis: Creating entity/ relationship diagram, data flow model, control flow model, the control and process specification; The data dictionary; Other classical analysis methods. System Design: Design concepts and principles: the design process: Design and software quality, design principles; Design concepts: Abstraction, refinement, modularity, software architecture, control hierarchy, structural partitioning, data structure, software procedure, information hiding; Effective modular design: Functional independence, Cohesion, Coupling; Design Heuristics for effective modularity; The design model; Design documentation.

Unit – 3	**Number of lectures = 08**	

Architectural Design: Software architecture, Data Design: Data modeling, data structures, databases and the data warehouse, Analyzing alternative Architectural Designs, architectural complexity; Mapping requirements Into software architecture; Transform flow, Transaction flow; Transform mapping: Refining the architectural design. Testing and maintenance: Software Testing Techniques, software testing fundamentals: objectives, principles, testability; Test case design, white box testing, basis path testing: Control structure testing: Black box testing, testing for specialized environments, architectures and applications.

Unit – 4	**Number of lectures = 10**	

Software Testing Strategies: Verification and validation, Unit testing, Integration testing,; Validation testing, alpha and beta testing; System testing: Recovery testing, security testing, stress testing, performance testing; The art of debugging, the debugging process debugging approaches. Software re-engineering , reverse engineering ,restructuring, forward engineering, Software Reliability and Quality Assurance :Quality concepts, Software quality assurance , SQA activities; Software reviews: cost impact of software defects, defect amplification and removal; formal technical reviews: The review meeting, review reporting and record keeping, review guidelines; Formal approaches to SQA; Statistical software quality assurance; software reliability: Measures of reliability and availability ,The ISO 9000 Quality standards: The ISO approach to quality assurance systems.

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/

Journal papers; Patents in the respective field.

13.Books Recommended

Text Book:

I. Software Engineering – A Practitioner's Approach, Roger S. Pressman, 2016, MGH.

Reference Books:

II. Fundamentals of software Engineering, Rajib Mall, PHI

III. Software Engineering by Ian sommerville, Pearson Edu, 5 edition, 1999, AW,

IV. Software Engineering – David Gustafson, 2002, T.M.H Software Engineering Fundamentals Oxford University

V. Ali Behforooz and Frederick J. Hudson 1995 JW&S, An Integrated Approach to software engineering by Pankaj jalote , 1991 Narosa,

SEMESTER III

Theory of Computation

<table>
<tr><td colspan="8">1. Name of the Department- CSE</td></tr>
<tr><td colspan="2">2. Course Name</td><td>Formal Language & Automata Theory</td><td>L</td><td colspan="2">T</td><td colspan="2">P</td></tr>
<tr><td colspan="2">3. Course Code</td><td></td><td>3</td><td colspan="2">0</td><td colspan="2">0</td></tr>
<tr><td colspan="3">4. Type of Course (use tick mark)</td><td>Core (√)</td><td colspan="2">PE()</td><td colspan="2">OE ()</td></tr>
<tr><td colspan="2">5. Pre-requisite (if any)</td><td></td><td>6. Frequency (use tick marks)</td><td>Even ()</td><td>Odd (✓)</td><td>Either Sem ()</td><td>Every Sem ()</td></tr>
<tr><td colspan="8">7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</td></tr>
<tr><td colspan="3">Lectures = 36</td><td>Tutorials = 0</td><td colspan="4">Practical = 0</td></tr>
<tr><td colspan="8">8. Course Description
• This course provides students a synopsis of latest trends in automotive industry used in evaluation of world. This includes understanding the basic principles of various hybrid and electric vehicles with importance, applications and limitations.</td></tr>
<tr><td colspan="8">9. Learning objectives:
i. Develop a formal notation for strings, languages and machines.
ii. Design finite automata to accept a set of strings of a language.
iii. Prove that a given language is regular and apply the closure properties of languages.
iv. Design context free grammars to generate strings from a context free language and convert them into normal forms.
v. Prove equivalence of languages accepted by Push Down Automata and languages generated by context free grammars
vi. Identify the hierarchy of formal languages, grammars and machines.
vii. Distinguish between computability and non-computability and Decidability and undecidability.</td></tr>
<tr><td colspan="8">10. Course Outcomes (COs):</td></tr>
<tr><td colspan="8">i. Write a formal notation for strings, languages and machines.</td></tr>
<tr><td colspan="8">ii. Design finite automata to accept a set of strings of a language.</td></tr>
<tr><td colspan="8">iii. For a given language determine whether the given language is regular or not.</td></tr>
<tr><td colspan="8">iv. Design context free grammars to generate strings of context free language .
v. Determine equivalence of languages accepted by Push Down Automata and languages generated by context free grammars</td></tr>
<tr><td colspan="8">vi. Write the hierarchy of formal languages, grammars and machines.</td></tr>
<tr><td colspan="8">vii. Distinguish between computability and non-computability and Decidability and undecidability.</td></tr>
<tr><td colspan="8">11. Unit wise detailed content</td></tr>
<tr><td colspan="2">Unit-1</td><td>Number of lectures = 9</td><td colspan="5">Title of the unit: Introduction</td></tr>
<tr><td colspan="8">Introduction: Alphabet, languages and grammars, productions and derivation, Chomsky hierarchy of languages, Regular languages and finite automata: Regular expressions and languages, deterministic finite automata (DFA) and equivalence with regular expressions, nondeterministic finite automata</td></tr>
</table>

(NFA) and equivalence with DFA, regular grammars and equivalence with finite automata, properties of regular languages, pumping lemma for regular languages, minimization of finite automata.		
Unit – 2	**Number of lectures = 9**	**Title of the unit: Context-free languages and pushdown automata**
Context-free languages and pushdown automata: Context-free grammars (CFG) and languages (CFL), Chomsky and Greibach normal forms, nondeterministic pushdown automata (PDA) and equivalence with CFG, parse trees, ambiguity in CFG, pumping lemma for context-free languages, deterministic pushdown automata, closure properties of CFLs.		
Unit – 3	**Number of lectures = 9**	**Title of the unit: Context-sensitive languages**
Context-sensitive languages: Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG.		
Unit – 4	**Number of lectures = 9**	**Title of the unit: Turing machines**
Turing machines: The basic model for Turing machines (TM), Turing-recognizable (recursively enumerable) and Turing-decidable (recursive) languages and their closure properties, variants of Turing machines, nondeterministic TMs and equivalence with deterministic TMs, unrestricted grammars and equivalence with Turing machines, TMs as enumerators. Undecidability: Church-Turing thesis, universal Turing machine, the universal and diagonalization languages, reduction between languages and Rice s theorem, undecidable problems about languages.		
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. Text Books Recommended		
• K.L.P Mishra, Theory Of Computer Science: Theory, Automata, And Computation, 3rd Edition, PHI,2006		
14. Reference Books Recommended		
• John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, Pearson Education Asia, , 3rd Edition,2016		
• Dexter C. Kozen, Automata and Computability, Undergraduate Texts in Computer Science, Springer.,2007		
• Michael Sipser, Introduction to the Theory of Computation, PWS Publishing.,3rd Edition ,2014		
• John Martin, Introduction to Languages and The Theory of Computation, Tata McGraw Hill.,4th Edition, 2010		

Semester III
MCA

<table>
<tr><td colspan="8">1. Name of the Department- Computer Science & Engineering</td></tr>
<tr><td colspan="2">2. Course Name</td><td colspan="2">Data Communication and Networks</td><td colspan="2">L</td><td>T</td><td>P</td></tr>
<tr><td colspan="2">3. Course Code</td><td colspan="2"></td><td colspan="2">3</td><td>0</td><td>0</td></tr>
<tr><td colspan="3">4. Type of Course (use tick mark)</td><td colspan="2">Core (✓)</td><td>PE()</td><td colspan="2">OE ()</td></tr>
<tr><td colspan="2">5. Pre-requisite (if any)</td><td>Computer Network</td><td colspan="2">6. Frequency (use tick marks)</td><td>Even ()</td><td>Odd (✓)</td><td>Either Sem ()</td><td>Every Sem ()</td></tr>
<tr><td colspan="8">7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</td></tr>
<tr><td colspan="3">Lectures = 36</td><td colspan="2">Tutorials = 0</td><td colspan="3">Practical = 0</td></tr>
<tr><td colspan="8">8. Course Description</td></tr>
<tr><td colspan="8">This course covers the data communication and computer network. The main contents are: LAN, WAN, MAN & wireless networks Laying architecture of networks, OSI model AM,FM and PM Multiple Access protocol-ALOHA network layer addressing Layer-4 protocol TCP & UDP TCP/IP, Protocols, Internet Protocol , Transmission control, User Datagram Protocol , IP Address classes, Subnet addressing ,Internet Email-SMTP, POP, IMAP, FTP NNTP, HTTP, SNMP, TELNET. Includes weekly laboratory</td></tr>
<tr><td colspan="8">9. Learning objectives:
To understand the concepts of data communication and to study the functions of different layers used in communication the data over network.
To introduce IEEE standards employed in computer networking. To make the students to get familiarized with different protocols and network components.</td></tr>
<tr><td colspan="8">10. Course Outcomes (COs):</td></tr>
<tr><td colspan="8">Understand the computer networks</td></tr>
<tr><td colspan="8">Design and analyze LAN</td></tr>
<tr><td colspan="8">Design and analyze WAN</td></tr>
<tr><td colspan="8">Design and analyze MAN</td></tr>
<tr><td colspan="8">Understand OSI, TCP/IP, HTTP etc</td></tr>
<tr><td colspan="8">11. Unit wise detailed content</td></tr>
<tr><td colspan="2">Unit-1</td><td colspan="2">Number of lectures = 9</td><td colspan="4"></td></tr>
<tr><td colspan="8">Introduction of Computer Networks, description of LAN, WAN, MAN & wireless networks Basic terminology of computer networks: - Bandwidth, physical and logical networks, Bridge, switch, HUB, Modem SCU/DSU
OSI Reference Model: Laying architecture of networks, OSI model, Function of each layer, Services and Protocols of each Layer.
Physical Layer: Representation of a bit on physical modem that is in wired network, optical network and wireless network, AM,FM and PM. Different types of media –twisted pair unshielded twisted pair, coaxial cable, optical Fiber cable and wireless.</td></tr>
<tr><td colspan="2">Unit – 2</td><td colspan="2">Number of lectures = 9</td><td colspan="4"></td></tr>
</table>

Data Link Layer: framing error control and flow control. Error detection & correction CRC block codes parity and check sum, elementary data link protocol, sliding window protocol, channel allocation problem-static and dynamic. Multiple Access protocol-ALOHA, CSMA/CU, Token ring, FDDI.

Network Layer: network layer addressing, network layer datagram, IP addressed Classes. Sub netting-Sub network, Subnet mask, Routing algorithm-optionally principle, Shortest path routing, hierarchical routing, Broadcast routing, Multicast routing, DHCP, Routing protocol.

Unit – 3	**Number of lectures = 9**	

Transport layer: Layer-4 protocol TCP & UDP. Three-way hand shakes open connection.

Introduction to Network Management: Remote Monitoring Techniques: Polling, Traps, Performance Management, Introduction to Network Operating System: Client- Server Infrastructure, WINDOWS nt/2000.

Unit – 4	**Number of lectures = 9**	

TCP/IP : Introduction History of TCP/IP, Protocols, Internet Protocol , Transmission control, User Datagram Protocol , IP Address classes, Subnet addressing ,Internet Email-SMTP, POP, IMAP, FTP NNTP, HTTP, SNMP, TELNET,

Application Layer: Domain name system, E-mail, File transfer protocol, HTTP, HTTPS, World Wide Web.

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

- Computer Networks: Tanenbaum, PHI, New Delhi, 12th Edition, 2020.

Reference Books

- Data Communication & Networking, Frouzen Tata McGraw Hill Publications, 8th Edition, 2020.
- Computer Networking: A Top-Down Approach, Kurose James F., Pearson Education; Ninth edition, 2020.
- Computer Networks - A System Approach, Elsevier; 14th edition, 2020.

Semester III
MCA

1. Name of the Department- Computer Science & Engineering						
2. Course Name	**Data Communication and Networks Lab**	**L**	**T**		**P**	
3. Course Code		**0**	**0**		**2**	
4. Type of Course (use tick mark)		**Core (✓)**	**PE()**		**OE ()**	
5. Pre-requisite (if any)	Computer Network Lab	**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 = 0		**Tutorials = 0**	**Practical = 24**			

8. Course Description

This course covers the data communication and computer network. The main contents are: LAN, WAN, MAN & wireless networks Laying architecture of networks, OSI model AM,FM and PM Multiple Access protocol-ALOHA network layer addressing Layer-4 protocol TCP & UDP TCP/IP, Protocols, Internet Protocol , Transmission control, User Datagram Protocol , IP Address classes, Subnet addressing ,Internet Email-SMTP, POP, IMAP, FTP NNTP, HTTP, SNMP, TELNET. Includes weekly laboratory

9. Learning objectives:

- Familiarize students with different Networks components such as switch, routers etc.
- Make them comfortable in socket programming and internet programming.

10.Course Outcomes (COs):

- Understand basic Network Commands.
- Understand the basic functioning of Switches and routers etc.
- Understand the functioning of different layers.

11. List of Experiments

1. Introduction to basic Linux networking commands. (Commands ipconfig and getmac)
2. Introduction to basic Linux networking commands. (Commands tracert and pathping)
3. Introduction to basic Linux networking commands. (Commands arp and ping, netstat, finger)
4. Implement bit stuffing.
5. Implement bit de-stuffing
6. Write a program for hamming code generation for error detection
7. Write a program for hamming code generation for error correction
8. Implement cyclic redundancy check (CRC).
9. Write a program for congestion control using the leaky bucket algorithm.
10. 13. Implementation of the link state routing protocols.
11. Implementation of LZW compression algorithms.
12. Implementation of LZW decompression algorithms.

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

http://vlabs.iitb.ac.in/vlabs-dev/labs_local/computer-networks/labs/explist.php
http://www.vlab.co.in/broad-area-electronics-and-communications

Semester III
MCA

<table>
<tr><td colspan="8">1. Name of the Department- Computer Science & Engineering</td></tr>
<tr><td>2.Course Name</td><td colspan="2">JAVA PROGRAMMING</td><td>L</td><td colspan="2">T</td><td colspan="2">P</td></tr>
<tr><td>3. Course Code</td><td colspan="2"></td><td>3</td><td colspan="2">0</td><td colspan="2">0</td></tr>
<tr><td colspan="2">4. Type of Course (use tick mark)</td><td>Core (✓)</td><td colspan="2">PE()</td><td colspan="2">OE ()</td></tr>
<tr><td>5. Pre-requisite (if any)</td><td>Basic knowledge of programming language e.g. C programming knowledge</td><td>6.Frequency (use tick marks)</td><td>Even ()</td><td>Odd (✓)</td><td>Either Sem ()</td><td>Every Sem ()</td></tr>
<tr><td colspan="8">7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</td></tr>
<tr><td colspan="3">Lectures = 36</td><td colspan="2">Tutorials = 0</td><td colspan="3">Practical = 0</td></tr>
<tr><td colspan="8">8. Course Description</td></tr>
<tr><td colspan="8">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 core concept of the java programming language.</td></tr>
<tr><td colspan="8">9. Learning objectives:
● To create, debug and run simple java programs in java SDK environment.
● To understand the fundamentals of object-oriented programming in java, which includes the definition of classes, methods and use of java libraries.
● To understand the application of java programming language in different technologies.</td></tr>
<tr><td colspan="8">10. Course Outcomes (COs):</td></tr>
<tr><td colspan="8">● Understanding the structure and model of the java programming language.</td></tr>
<tr><td colspan="8">● Using java programming language to develop various applications.</td></tr>
<tr><td colspan="8">● Develop software using java programming language.</td></tr>
<tr><td colspan="8"></td></tr>
<tr><td colspan="8">11. Unit wise detailed content</td></tr>
<tr><td>Unit-1</td><td colspan="2">Number of lectures = 10</td><td colspan="5"></td></tr>
<tr><td colspan="8">Importance and features of Java: Language Construct of java including Keywords,constants,Programming language Types and Paradigms, Computer Programming Hierarchy, How Computer Architecture Affects a Language? Why Java? Flavors of Java, Java Designing Goal, Role of Java Programmer in Industry, Features of Java Language, JVM –The heart of Java, Java’s Magic Bytecode.Installing Java, Java Program Development, Java Source File Structure,Compilation, Executions.Lexical Tokens, Identifiers, Keywords, Literals, Comments,Primitive Datatypes, Operators Assignments.
Introducing classes, objects and methods: defining Class Fundamentals, Object & Object reference, Object Life time & Garbage Collection, Creating and Operating Objects , Constructor & initialization code block, Access Control, Modifiers, methods Nested , Inner</td></tr>
</table>

Class &Anonymous Classes ,Abstract Class & 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 & Methods, Design of Accessors and Mutator Methods Cloning Objects, shallow and deep cloning, Generic Class Types.

Unit – 2	**Number of lectures = 8**	

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.

Unit – 3	**Number of lectures = 9**	

Exception Handling: The Idea behind Exception,Exceptions & Errors,Types of Exception, Control Flow In Exceptions, JVM reaction to Exceptions,Use of try, catch, finally, throw, throws in Exception Handling,In-built and User Defined Exceptions, Checked and Un-Checked Exceptions.

Package: Organizing Classes and Interfaces in Packages, Package as Access Protection , Defining Package ,CLASSPATH Setting for Packages , Making JAR Files for Library Packages Import and Static Import Naming Convention For Packages.

Unit – 4	**Number of lectures = 9**	

Array & String: Defining an Array, Initializing & Accessing Array, Multi –Dimensional Array, Operation on String, Mutable & Immutable String, Using Collection Bases Loop for String, Tokenizing a String, Creating Strings using StringBuffer .

A Collection of Useful Classes: Utility Methods for Arrays ,Observable and Observer Objects , Date & Times ,Using Scanner Regular Expression, Input/output Operation in Java(java.io Package),Streams and the new I/O Capabilities ,Understanding Streams, The Classes for Input and Output, The Standard Streams, Working with File Object, File I/O Basics, Reading and Writing to Files, Buffer and Buffer Management, Read/Write Operations with File Channel, Serializing Objects .

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

- Java, Herbert Schildt. "The Complete Reference." Complete Reference Series) 10th

Edition| New York: McGraw-Hill Education (2017).

Reference Books

- SAMANTA, DEBASIS. Object-oriented Programming with C++ and Java. PHI Learning Pvt. Ltd., 2006.
- https://cse.iitkgp.ac.in/~dsamanta/java/index.htm, https://nptel.ac.in/courses/106/105/106105191/
- E. Balaguruswamy, "Programming with Java: A Primer", McGraw-Hill; Sixth edition, 2019.

Semester III
MCA

<table>
<tr><td colspan="8">1. Name of the Department- Computer Science & Engineering</td></tr>
<tr><td>2. Course Name</td><td colspan="2">Java Programming lab</td><td colspan="2">L</td><td colspan="2">T</td><td>P</td></tr>
<tr><td>3. Course Code</td><td colspan="2"></td><td colspan="2">0</td><td colspan="2">0</td><td>2</td></tr>
<tr><td colspan="3">4. Type of Course (use tick mark)</td><td colspan="2">Core (✔)</td><td colspan="2">PE()</td><td>OE ()</td></tr>
<tr><td>5. Pre-requisite (if any)</td><td colspan="2">Knowledge of C</td><td>6. Frequency (use tick marks)</td><td>Even ()</td><td>Odd (✔)</td><td>Either Sem ()</td><td>Every Sem ()</td></tr>
<tr><td colspan="8">7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</td></tr>
<tr><td colspan="3">Lectures = 0</td><td colspan="2">Tutorials = 0</td><td colspan="3">Practical = 24</td></tr>
<tr><td colspan="8">8.Course Description
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 core concept of the java programming language.</td></tr>
<tr><td colspan="8">9. Learning objectives:
● To create, debug and run simple java programs in java SDK environment.
● To understand the fundamentals of object-oriented programming in java, which includes the definition of classes, methods and use of java libraries.
● To understand the application of java programming language in different technologies.</td></tr>
<tr><td colspan="8">10.Course Outcomes (COs):</td></tr>
<tr><td colspan="8">● Understanding the structure and model of the java programming language.</td></tr>
<tr><td colspan="8">● Using java programming language to develop various applications.</td></tr>
<tr><td colspan="8">● Develop software using java programming language.</td></tr>
<tr><td colspan="8">11. List of Experiments</td></tr>
<tr><td colspan="8">1. Write a program to swap two values using object reference. Your program should have a swap function.
2. Write an application that accepts one command line argument; display the line of reporting if number is even or odd.
3. 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.
4. Write a program to show the concept of Constructors.
5. WAP that shows passing object as parameter.
6. WAP that illustrates method overriding.
7. WAP to illustrate dynamic polymorphism.
8. Write a program to show the concept of method overloading.
9. Write a program to show the concept of Inheritance.
10. WAP illustrating a super class variable a referencing as sub class object.
11. WAP illustrating all uses of super keywords.
12. Write an application that shows the usage of try, catch, throws and finally.</td></tr>
</table>

13. Write an application that shows how to create a user-defined exception.
14. Create a customized exception and also make use of all the 5 exception keywords.
15. Write a program to show the concept of packages.

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

https://www.codecademy.com/learn/learn-java
https://www.learnjavaonline.org/

MCA

Semester III

<table>
<tr><td colspan="8">1. Name of the Department- Computer Science & Engineering</td></tr>
<tr><td colspan="2">2. Course Name</td><td colspan="2">Mobile Computing</td><td colspan="2">L</td><td colspan="2">T</td><td>P</td></tr>
<tr><td colspan="2">3. Course Code</td><td colspan="2"></td><td colspan="2">3</td><td colspan="2">1</td><td></td></tr>
<tr><td colspan="3">4. Type of Course (use tick mark)</td><td colspan="2">Core ()</td><td colspan="2">PE(✓)</td><td colspan="2">OE ()</td></tr>
<tr><td colspan="2">5. Pre-requisite (if any)</td><td>MWC</td><td colspan="2">6. Frequency (use tick marks)</td><td>Even ()</td><td>Odd (✓)</td><td>Either Sem ()</td><td>Every Sem ()</td></tr>
<tr><td colspan="9">7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</td></tr>
<tr><td colspan="3">Lectures = 36</td><td colspan="3">Tutorials = 0</td><td colspan="3">Practical = 0</td></tr>
<tr><td colspan="9">8. Course Description</td></tr>
<tr><td colspan="9">This course will cover various topics of mobile computing, networking, and systems, including but not limited to: applications of smartphones, cellular networks, embedded sensor systems, localization systems, energy efficiency of mobile devices, wearable and vehicular mobile systems, mobile security</td></tr>
<tr><td colspan="9">9. Learning objectives:
To impart fundamental concepts in the area of mobile computing, to provide a computer systems perspective on the converging areas of wireless networking, embedded systems, and software, and to introduce selected topics of current research interest in the field.</td></tr>
<tr><td colspan="9">10. Course Outcomes (COs):</td></tr>
<tr><td colspan="9">• Understand fundamentals of wireless communications.</td></tr>
<tr><td colspan="9">• Analyze security, energy efficiency, mobility, scalability, and their unique characteristics in wireless networks.</td></tr>
<tr><td colspan="9">• Demonstrate basic skills for cellular networks design.</td></tr>
<tr><td colspan="9">• Apply knowledge of TCP/IP extensions for mobile and wireless networking.</td></tr>
<tr><td colspan="9">11. Unit wise detailed content</td></tr>
<tr><td colspan="2">Unit-1</td><td colspan="2">Number of lectures = 09</td><td colspan="5"></td></tr>
<tr><td colspan="9">Cellular Mobile Wireless Networks: Systems and Design Fundamentals, Propagation Models Description of Cellular system, Frequency Reuse, Co channel and Adjacent channel interference, Propagation Models for Wireless Networks, Multipart Effects in Mobile Communication, Models for Multipart Reception Evolution of Modern Mobile Wireless Communication System.</td></tr>
<tr><td colspan="2">Unit – 2</td><td colspan="2">Number of lectures = 09</td><td colspan="5"></td></tr>
<tr><td colspan="9">First Generation Wireless Networks, Second Generation (2G) Wireless Cellular Networks, Major 2G standards, 2.5G Wireless Networks, Third Generation 3G Wireless Networks, Wireless Local Area Networks (WLANs), All-IP Network: Vision for 4GIssues in Mobile computing, Wireless Multiple Access protocols , channel Allocation.</td></tr>
<tr><td colspan="2">Unit – 3</td><td colspan="2">Number of lectures = 09</td><td colspan="5"></td></tr>
<tr><td colspan="9">Data management issues: mobility, wireless communication and portability, data replication Schemes, basic concept of multihopping, Adaptive Clustering for mobile Network, Multicluster</td></tr>
</table>

Architecture. Location Management: Introduction, Location Based Services, Automatically Locating Mobile Users		
Unit – 4	**Number of lectures = 09**	
Locating and Organizing Services, Is Use and future directions, mobile IP, Comparison of TCP wireless. Transaction management: Introduction, Data Dissemination, Cache Consistency, Mobile transaction processing, mobile database research directions, Security fault tolerance for mobile N/W.		
12. Brief Description of self-learning / E-learning component https://elearning.sgtuniversity.ac.in/course-category/MOBILE COMPUTING		
13. Books Recommended		
Text Books 1. Mobile Communications, Schiller, Pearson Education India; New edition.		
Reference Books		
1. Mobile Computing, Shambhu Upadhyaya, Abhjeet Chaudhary, Keviven Kwiat, Mark Weises , Kliuwer Acadmic Publishers.		
2. Principles of Mobile computing, UIWE Hansmann, Other Merk , Martin-S-Nickious, Thomas Stohe, Springer international Edition.		
3. Mobile Computing, Sipra DasBit, Biplab K. Sikdar, PHI, New Edition.		

Semester IV
MCA

<table>
<tr><td colspan="8">1. Name of the Department- Computer Science & Engineering</td></tr>
<tr><td colspan="2">2. Course Name</td><td>Cloud Computing</td><td colspan="2">L</td><td colspan="2">T</td><td>P</td></tr>
<tr><td colspan="2">3. Course Code</td><td></td><td colspan="2">3</td><td colspan="2">0</td><td>0</td></tr>
<tr><td colspan="3">4. Type of Course (use tick mark)</td><td colspan="2">Core (✓)</td><td colspan="2">PE()</td><td>OE ()</td></tr>
<tr><td colspan="2">5. Pre-requisite (if any)</td><td>Computer Network, Operating System, Algorithms</td><td colspan="2">6. Frequency (use tick marks)</td><td>Even (✓)</td><td>Odd ()</td><td>Either Sem ()</td><td>Every Sem ()</td></tr>
<tr><td colspan="8">7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</td></tr>
<tr><td colspan="3">Lectures = 36</td><td colspan="3">Tutorials = 0</td><td colspan="2">Practical = 0</td></tr>
<tr><td colspan="8">8. Course Description</td></tr>
<tr><td colspan="8">This course covers a series of current cloud computing technologies, including technologies for Infrastructure as a Service, Platform as a Service, Software as a Service, and Physical Systems as a Service. The course is also highly project oriented, involving hand-on exploration of existing technologies as well as development of new technologies.</td></tr>
<tr><td colspan="8">9. Learning objectives:
• To introduce Cloud Computing Technologies as used in Industry.
• To give understanding Service Models & Deployment Model in Cloud Computing.
• To familiarize the Concept of Virtualisation & learn the use cases of Cloud Computing with the help of Case Study.</td></tr>
<tr><td colspan="8">10. Course Outcomes (COs):</td></tr>
<tr><td colspan="8">● Applying key comparative methodologies to assess the comparative advantages and disadvantages of public vs. private computing clouds</td></tr>
<tr><td colspan="8">● Applying relevant methods to assess the important security and sustainability challenges involved in adopting various cloud architectures</td></tr>
<tr><td colspan="8">● Applying Cloud Computing to Industry Use Cases</td></tr>
<tr><td colspan="8">11. Unit wise detailed content</td></tr>
<tr><td colspan="2">Unit-1</td><td>Number of lectures = 09</td><td colspan="5"></td></tr>
<tr><td colspan="8">Introduction to Cloud Computing, History of Cloud Computing, Cloud service providers, Pros and Cons of Cloud Computing, Benefits of Cloud Computing, Cloud computing vs. Cluster computing vs. Grid computing.</td></tr>
<tr><td colspan="2">Unit – 2</td><td>Number of lectures = 09</td><td colspan="5"></td></tr>
<tr><td colspan="8">Cloud Computing Architecture, Service Models (XaaS), Infrastructure as a Service(IaaS), Platform as a Service(PaaS), Software as a Service(SaaS). Application of Service Models.</td></tr>
<tr><td colspan="2">Unit – 3</td><td>Number of lectures = 9</td><td colspan="5"></td></tr>
</table>

Deployment Models, Public cloud, Private cloud, Hybrid cloud, Community cloud,Concept of Virtualisation, Cloud security, Cloud Economics		
Unit – 4	**Number of lectures = 09**	
Case Study on Open Source & Commercial Clouds: Eucalyptus, Microsoft Azure, Amazon EC2.		
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 ● Cloud Computing (Wind) by Dr. Kumar Saurabh, 2nd Edison, Wiley India		
Reference Books		
● Cloud Computing: Principles and Paradigms, Editors: Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Wile, 2011		
● Cloud Computing: Principles, Systems and Applications, Editors: Nikos Antonopoulos, Lee Gillam, Springer, 2012		

Semester III
MCA

<table>
<tr><td colspan="8">1. Name of the Department- Computer Science & Engineering</td></tr>
<tr><td colspan="2">2. Course Name</td><td>Cloud Computing Lab</td><td>L</td><td colspan="2">T</td><td colspan="2">P</td></tr>
<tr><td colspan="2">3. Course Code</td><td></td><td>0</td><td colspan="2">0</td><td colspan="2">2</td></tr>
<tr><td colspan="3">4. Type of Course (use tick mark)</td><td>Core (✓)</td><td colspan="2">PE()</td><td colspan="2">OE ()</td></tr>
<tr><td colspan="2">5. Pre-requisite (if any)</td><td>Computer Network, Operating System, Algorithms</td><td>6. Frequency (use tick marks)</td><td>Even (✓)</td><td>Odd ()</td><td>Either Sem ()</td><td>Every Sem ()</td></tr>
<tr><td colspan="8">7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</td></tr>
<tr><td colspan="3">Lectures = 0</td><td colspan="2">Tutorials = 0</td><td colspan="3">Practical = 24</td></tr>
<tr><td colspan="8">8. Course Description</td></tr>
<tr><td colspan="8">9. Learning objectives:
● To analyze, design and provide optimal solution for Computer Science & Engineering and multidisciplinary problems.
● To pursue research by applying knowledge of mathematics and fundamentals of computer science.
● To exhibit recently emerging technical skills and adapt to current trends by engaging in lifelong learning.</td></tr>
<tr><td colspan="8">10. Course Outcomes (COs):</td></tr>
<tr><td colspan="8">● Understand the significance of problem analysis</td></tr>
<tr><td colspan="8">● Design and development of solutions to very complex engineering problems</td></tr>
<tr><td colspan="8">● Enabling modern tools usage.</td></tr>
<tr><td colspan="8">● Understand the recent trends in computation and sustainability</td></tr>
<tr><td colspan="8">● Design & Analyze cloud computing use cases and applicability.</td></tr>
<tr><td colspan="8">11. List of Experiments</td></tr>
<tr><td colspan="8">1. To understand the Industry Use-Cases of Cloud Computing.
2. Creating a Warehouse Application in SalesForce.com.
3. Creating an Application in SalesForce.com using Apex programming Language.
4. Implementation of SOAP Web services in C#/JAVA Applications.
5. Deploying & Testing the Web Service.
6. Implementation of Para-Virtualization using VM Ware‘s Workstation/ Oracle‘s Virtual Box.
7. Installation of Guest OS.
8. Installation of Hadoop.
9. Configuration of Hadoop.
10. Understanding Map Reduce.
11. Case Study: Facebook.
12. Case Study: Google App Engine.
13. Case Study: AWS.
14. Case Study: Netflix.</td></tr>
</table>

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

The students will be encouraged to learn using Virtual Link.

i. https://hadoop.apache.org/
ii. https://aws.amazon.com/
iii. https://cloud.google.com/appengine

Semester IV
MCA

<table>
<tr><td colspan="8">Name of the Department- Computer Science and Engineering</td></tr>
<tr><td>Course Name</td><td colspan="2">Computer Graphics & Multimedia</td><td colspan="2">L</td><td colspan="2">T</td><td>P</td></tr>
<tr><td>Course Code</td><td colspan="2"></td><td colspan="2">3</td><td colspan="2">0</td><td>0</td></tr>
<tr><td colspan="2">Type of Course (use tick mark)</td><td colspan="2">Core (√)</td><td colspan="2">PE()</td><td colspan="2">OE ()</td></tr>
<tr><td>Pre-requisite (if any)</td><td colspan="2"></td><td>Frequency (use tick marks)</td><td>Even (√)</td><td>Odd ()</td><td>Either Sem ()</td><td>Every Sem ()</td></tr>
<tr><td colspan="8">Total Number of Lectures, Tutorials, Practical (assuming 12weeks of one semester)</td></tr>
<tr><td colspan="3">Lectures = 36</td><td colspan="2">Tutorials = 0</td><td colspan="3">Practical = 0</td></tr>
<tr><td colspan="8">Course Description</td></tr>
<tr><td colspan="8">Computer Graphics is a study of the hardware and software principles of interactive raster graphics. Students will use a standard computer graphics API to reinforce concepts and study fundamental computer graphics algorithms.</td></tr>
<tr><td colspan="8">Learning objectives:
● To provide a comprehensive introduction to computer graphics leading to the ability to understand contemporary terminology, progress, issues, and trends.
● To understand computer graphics techniques (2-D/3-D), focusing on 3D modelling, image synthesis, and rendering.
● Introduce geometric transformations, geometric algorithms, software systems (OpenGL), 3D object models (surface, volume and implicit), visible surface algorithms, image synthesis, shading and mapping, ray tracing, radiosity, global illumination, photon mapping, and anti-aliasing.
● To explore the interdisciplinary nature of computer graphics which is emphasized in the wide variety of examples and applications.</td></tr>
<tr><td colspan="8">Course Outcomes (COs):</td></tr>
<tr><td colspan="8"></td></tr>
<tr><td colspan="8">● To develop a facility with the relevant mathematics of computer graphics, e.g., 3D rotations using both vector algebra, geometrical transformations and projections using homogeneous co-ordinations.
● Apply principles and techniques of computer graphics, e.g., the graphics pipeline, and Bresenham algorithm for speedy line and circle generation.
● Apply computer graphics concepts in the development of computer games, information visualization, and business applications.</td></tr>
<tr><td colspan="8"></td></tr>
<tr><td colspan="8"></td></tr>
<tr><td colspan="8">Unit wise detailed content</td></tr>
<tr><td>Unit-1</td><td colspan="2">Number of lectures = 08</td><td colspan="5">Title of the unit: Introduction to Line generation</td></tr>
<tr><td colspan="8">Types of computer graphics, Graphic Displays- Random scan displays, Raster scan displays, Frame</td></tr>
</table>

buffer and video controller, Points and lines, Line drawing algorithms, Circle generating algorithms, Midpoint circle generating algorithm, and parallel version of these algorithms.

Unit – 2	**Number of lectures = 10**	**Title of the unit: Transformations**

Basic transformation, Matrix representations and homogenous coordinates, Composite transformations, Reflections and shearing. Windowing and Clipping: Viewing pipeline, Viewing transformations, 2-D Clipping algorithms-Line clipping algorithms such as Cohen Sutherland line clipping algorithm, Liang Barsky algorithm, Line clipping against nonrectangular clip windows; Polygon clipping – Sutherland Hodgeman polygon clipping, Weiler and Atherton polygon clipping, Curve clipping, Text clipping.

Unit – 3	**Number of lectures = 08**	**Title of the unit: 3 Dimensional**

3-D geometric primitives, 3-D Object representation, 3-D Transformation, 3-D viewing, projections, 3-D Clipping. Quadric surfaces, Spheres, Ellipsoid, Blobby objects, introductory concepts of Spline, Bspline and Bezier curves and surfaces.

Unit – 4	**Number of lectures = 10**	**Title of the unit: Curves and Surfaces**

Back Face Detection algorithm, Depth buffer method, A- buffer method, Scan line method, basic illumination models – Ambient light, Diffuse reflection, Specular reflection and Phong model, Combined approach, Warn model, Intensity Attenuation, Color consideration, Transparency and Shadows.

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/
Journal papers; Patents in the respective field.

Books Recommended

- Computer Graphics C Version - Donald Hearn and M Pauline Baker, Pearson Education
- Computer Graphics - Amrendra N Sinha and Arun D Udai, TMH Publications
- Computer Graphics: A Programming Approach - Steven Harrington, TMH Publications
- Procedural Elements of Computer Graphics - Rogers, McGraw Hill

Semester IV
MCA

<table>
<tr><td colspan="8">1. Name of the Department: CSE</td></tr>
<tr><td colspan="2">2. Course Name</td><td>ADA</td><td colspan="2">L</td><td colspan="2">T</td><td>P</td></tr>
<tr><td colspan="2">3. Course Code</td><td></td><td colspan="2">3</td><td colspan="2">0</td><td>0</td></tr>
<tr><td colspan="3">4. Type of Course (use tick mark)</td><td colspan="2">Core (√)</td><td colspan="2">PE()</td><td>OE()</td></tr>
<tr><td colspan="2">5. Pre-requisite (if any)</td><td>Fundamental of C</td><td colspan="2">6. Frequency (use tick marks)</td><td>Even (√)</td><td>Odd ()</td><td>Either Sem ()</td><td>Every Sem ()</td></tr>
<tr><td colspan="9">7. Total Number of Lectures, Tutorials, Practical (assuming 14 weeks of one semester)</td></tr>
<tr><td colspan="3">Lectures = 36</td><td colspan="3">Tutorials = 0</td><td colspan="3">Practical = 0</td></tr>
<tr><td colspan="9">8. Brief Syllabus
• Introduction
• Advance Data Structures
• Dynamic Programming
• Algebraic computation</td></tr>
<tr><td colspan="9">9. Learning objectives:
i. Existing algorithm and develop efficient algorithms for simple computational tasks.
ii. Reasoning about the correctness of the algorithm
iii. Behaviours' of algorithms and the notion of tractable and intractable problems.</td></tr>
<tr><td colspan="9">10. Course Outcomes:

At the end of the course student will be able to:
i. Analyse algorithms and determine efficiency of algorithm.
ii. Understand advanced abstract data type (ADT), data structures and their implementations
iii. Design algorithms using the brute force, greedy, divide and conquer, branch and bound etc. methodologies.
iv. Prove problems of P, NP, or NP-Complete.
v. Develop and implement learned/new algorithm using appropriate techniques to solve problems.</td></tr>
<tr><td colspan="9">11. Unit wise detailed content</td></tr>
<tr><td colspan="2">Unit-1</td><td>Number of lectures = 10</td><td colspan="6">Title of the unit: Introduction</td></tr>
<tr><td colspan="9">Algorithms, Analyzing algorithms, Complexity of algorithms, Growth of functions, Performance measurements, Sorting and order Statistics - Shell sort, Quick sort, Merge sort, Heap sort, Comparison of sorting algorithms, Sorting in linear time.</td></tr>
<tr><td colspan="2">Unit – 2</td><td>Number of lectures =10</td><td colspan="6">Title of the unit: Advanced Data Structures</td></tr>
<tr><td colspan="9">Advanced Data Structures: Red-Black trees, B – trees, Binomial Heaps, Fibonacci Heaps.

Divide and Conquer with examples such as Sorting, Matrix Multiplication, Convex hull and Searching. Greedy methods with examples Huffman Coding, Knapsack, Minimum Spanning trees – Prim's and Kruskal's algorithms, Single source shortest paths - Dijkstra's and Bellman Ford</td></tr>
</table>

algorithms.		
Unit – 3	**Number of lectures = 8**	**Title of the unit: Dynamic Programming**
Dynamic programming with examples such as Knapsack, All pair shortest paths –Warshal's and Floyd's algorithms, Resource allocation problem. Backtracking, Branch and Bound with examples such as Travelling Salesman Problem, Graph Coloring, n-Queen Problem, Hamiltonian Cycles and Sum of subsets.		
Unit – 4	**Number of lectures = 8**	**Title of the unit: Computation**
Algebraic computation, String Matching, Theory of NP-completeness, Approximation algorithms and Randomized algorithms.		
12. Brief Description of self learning / E-learning component Online Video Lectures on Analysis and design of algorithms Practice of Algorithims		
13. Books Recommended (1Text Books + 4 Reference Books)		
• Thomas H. Coreman, Charles E. Leiserson and Ronald L. Rivest, "Introduction to Algorithms", Printice Hall of India.		
• RCT Lee, SS Tseng, RC Chang and YT Tsai, "Introduction to the Design and Analysis of Algorithms", Mc Graw Hill, 2005		
REFERENCE BOOKS		
• E. Horowitz & S Sahni, "Fundamentals of Computer Algorithms",		
• Berman, Paul," Algorithms", Cengage Learning.		
• Aho, Hopcraft, Ullman, "The Design and Analysis of Computer Algorithms" Pearson Education, 2008.		

Semester IV
MCA

<table>
<tr><td colspan="8">1. Name of the Department: CSE</td></tr>
<tr><td colspan="2">2.Course Name</td><td>Artificial Intelligence</td><td colspan="2">L</td><td>T</td><td colspan="2">P</td></tr>
<tr><td colspan="2">3. Course Code</td><td></td><td colspan="2">3</td><td>0</td><td colspan="2">2</td></tr>
<tr><td colspan="3">4. Type of Course (use tick mark)</td><td colspan="2">Core (√)</td><td>PE()</td><td colspan="2">OE()</td></tr>
<tr><td colspan="2">5. Pre-requisite (if any)</td><td>C++</td><td>6. Frequency (use tick marks)</td><td>Even (√)</td><td>Odd ()</td><td>Either Sem ()</td><td>Every Sem ()</td></tr>
<tr><td colspan="8">7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</td></tr>
<tr><td colspan="3">Lectures = 36</td><td colspan="2">Tutorials = 0</td><td colspan="3">Practical = 24</td></tr>
<tr><td colspan="8">8. Brief Syllabus:
• Introduction & Searching Strategies
• Knowledge Representation & Reasoning
• Machine Learning & Pattern Recognition
• Pattern Recognition</td></tr>
<tr><td colspan="8">9. Learning objectives:

i. learn and possess a firm grounding in the existing techniques and component areas of Artificial Intelligence
ii. Apply this knowledge to the development of Artificial Intelligent Systems and to the exploration of research problems.</td></tr>
<tr><td colspan="8">10. Course Outcomes:

i. Understand the principles of problem solving and be able to apply them successfully
ii. Be familiar with techniques for computer-based representation and manipulation of complex information, knowledge, and uncertainty
iii. Gain awareness of several advanced AI applications and topics such as intelligent agents, planning and scheduling, ma-chine learning, etc.
iv. Understand the principles of problem solving and be able to apply them successfully</td></tr>
<tr><td colspan="8">11. Unit wise detailed content</td></tr>
<tr><td colspan="2">Unit-1</td><td>Number of lectures = 9</td><td colspan="5">Introduction</td></tr>
<tr><td colspan="8">Introduction to Artificial Intelligence, Foundations and History of Artificial Intelligence, Applications of Artificial Intelligence, Intelligent Agents, Structure of Intelligent Agents. Computer vision, Natural Language Possessing.Searching for solutions, Uniformed search strategies, Informed search strategies, Local search algorithms and optimistic problems, Adversarial Search, Search for games, Alpha - Beta pruning.</td></tr>
<tr><td colspan="2">Unit - II</td><td>Number of lectures = 9</td><td colspan="5">Knowledge Representation & Reasoning</td></tr>
</table>

Propositional logic, Theory of first order logic, Inference in First order logic, Forward & Backward chaining, Resolution, Probabilistic reasoning, Utility theory, Hidden Markov Models (HMM), Bayesian Networks.		
Unit – III	**Number of lectures = 9**	**Machine Learning**
Supervised and unsupervised learning, Decision trees, Statistical learning models, Learning with complete data - Naive Bayes models, Learning with hidden data – EM algorithm, Reinforcement learning.		
Unit – IV	**Number of lectures = 9**	**Pattern Recognition**
Introduction, Design principles of pattern recognition system, Statistical Pattern recognition, Parameter estimation methods - Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA).		

12. 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.

13. Books Recommended (2 Text Books + 2 Reference Books)

- Artificial Intelligence – A Modern Approach - **St**uart Russell and Peter Norvig, Pearson Education.
- Artificial Intelligence - Elaine Rich and Kevin Knight, McGraw-Hill
- Introduction to Artificial Intelligence - E Charniak and D McDermott, Pearson Education
- Artificial Intelligence and Expert Systems - Dan W. Patterson, Prentice Hall of India

Semester IV
MCA

<table>
<tr><td colspan="8">2. Name of the Department- Computer Science & Engineering</td></tr>
<tr><td>2. Course Name</td><td colspan="2">Artificial Intelligence lab</td><td colspan="2">L</td><td colspan="2">T</td><td>P</td></tr>
<tr><td>3. Course Code</td><td colspan="2"></td><td colspan="2">0</td><td colspan="2">0</td><td>2</td></tr>
<tr><td colspan="3">4. Type of Course (use tick mark)</td><td colspan="2">Core (✔)</td><td colspan="2">PE()</td><td>OE ()</td></tr>
<tr><td>5. Pre-requisite (if any)</td><td colspan="2">Knowledge of C++</td><td>6. Frequency (use tick marks)</td><td>Even (✔)</td><td>Odd ()</td><td>Either Sem ()</td><td>Every Sem ()</td></tr>
<tr><td colspan="8">7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</td></tr>
<tr><td colspan="3">Lectures = 0</td><td colspan="2">Tutorials = 0</td><td colspan="3">Practical = 24</td></tr>
<tr><td colspan="8">8.Course Description
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 core concept of the java programming language.</td></tr>
<tr><td colspan="8">9. Learning objectives:
• learn and possess a firm grounding in the existing techniques and component areas of Artificial Intelligence
• Apply this knowledge to the development of Artificial Intelligent Systems and to the exploration of research problems.</td></tr>
<tr><td colspan="8">10.Course Outcomes (COs):</td></tr>
<tr><td colspan="8">• Understand the principles of problem solving and be able to apply them successfully</td></tr>
<tr><td colspan="8">• Be familiar with techniques for computer-based representation and manipulation of complex information, knowledge, and uncertainty</td></tr>
<tr><td colspan="8">• Gain awareness of several advanced AI applications and topics such as intelligent agents, planning and scheduling, ma-chine learning, etc.</td></tr>
<tr><td colspan="8">11. List of Experiments</td></tr>
<tr><td colspan="8">• Study of PROLOG.
• Write a program to solve 8 queens problem.
• Solve any problem using depth first search.
• Solve any problem using best first search.
• Solve 8-puzzle problem using best first search
• Solve Robot (traversal) problem using means End Analysis.
• Solve traveling salesman problem.</td></tr>
</table>

Semester V

MCA

<table>
<tr><td colspan="8">1. Name of the Department- Computer Science & Engineering</td></tr>
<tr><td colspan="2">2. Course Name</td><td>Advanced Web Technology</td><td>L</td><td colspan="2">T</td><td colspan="2">P</td></tr>
<tr><td colspan="2">3. Course Code</td><td></td><td>3</td><td colspan="2">0</td><td colspan="2">0</td></tr>
<tr><td colspan="2">4. Type of Course (use tick mark)</td><td colspan="2">Core (✓)</td><td colspan="2">PE()</td><td colspan="2">OE ()</td></tr>
<tr><td colspan="2">5. Pre-requisite (if any)</td><td>Computer Fundamentals</td><td>6. Frequency (use tick marks)</td><td>Even</td><td>Odd (✓)</td><td>Either Sem ()</td><td>Every Sem ()</td></tr>
<tr><td colspan="8">7. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</td></tr>
<tr><td colspan="3">Lectures = 36</td><td colspan="2">Tutorials = 0</td><td colspan="3">Practical = 0</td></tr>
<tr><td colspan="8">8. Course Description</td></tr>
<tr><td colspan="8">To help the students to understand the concept of HTML, CSS, Java script, PHP and MYSQL.</td></tr>
<tr><td colspan="8">9. Learning objectives:
Upon successful completion of the course in this discipline the student will be able to develop a complete dynamic website with data base as backend.
● To learn fundamental language of internet i.e. HTML and cascading style sheets.
● To learn basics of client side JavaScript and server side programming constructs.
● To design multimedia pages over web.</td></tr>
<tr><td colspan="8">10. Course Outcomes (COs):</td></tr>
<tr><td colspan="8">● How to design and develop a dynamic website.
● Basic knowledge of web services which are useful for the same.
● Acquainted with the difference between client side and server side scripting.
● Import multimedia pages over web.</td></tr>
<tr><td colspan="8"></td></tr>
<tr><td colspan="8">11. Unit wise detailed content</td></tr>
<tr><td colspan="2">Unit-1</td><td>Number of lectures = 10</td><td colspan="5"></td></tr>
</table>

HTML :- Basics of HTML, formatting and fonts, hyperlink, tables, images, forms, XHTML, Meta tags, Browser architecture and Web site structure. Overview and features of HTML5.

Style Sheets: Introduction to CSS, Need for CSS, basic syntax and structure using CSS, background images, colors and properties, manipulating texts, using fonts, borders and boxes, margins, padding lists, positioning using CSS.

Unit – 2	**Number of lectures = 9**	

Java Script:-

Introduction, Client-Side JavaScript, Server-Side JavaScript, JavaScript Objects, JavaScript Security, Operators, Statements, Document and its associated objects, Events and Event Handlers, Core JavaScript (Properties and Methods of Each)

Unit – 3	**Number of lectures = 08**	

PHP (Hypertext Preprocessor): Introduction, syntax, variables, strings, operators, if-else, loop, switch, array, function, form, mail, file upload, session, error, exception, filter, PHP-ODBC.

Unit – 4	**Number of lectures = 9**	

MYSQL: Introduction to Database and MYSQL, RDBMS-Understanding Tables, Records & Fields, SQL language, MYSQL queries.

Working with MYSQL Admin: Working with PHP My admin, data types, creating Database and tables, dropping Database and tables, adding fields, selecting table, Altering fields properties.

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

- PHP for the Web: Visual QuickStart Guide, Ullman, Pearson Education; Fifth edition, 2017.

Reference Books

- Web Technologies: HTML, JAVASCRIPT, PHP, JAVA, JSP, ASP.NET, XML and Ajax, Black Book: HTML, Javascript, PHP, Java, Jsp, XML and Ajax, Black Book, Kogent Learning Solutions Inc., Dreamtech Press; 1 edition, 2009.
- Mastering HTML, CSS & Javascript Web Publishing, Laura Lemay, BPB Publications; First edition, 2016.
- Beginning HTML5 with CSS3, Christopher Murphy , Apress publisher,1st ed. Edition, 2012.

Semester V

MCA

1. Name of the Department- Computer Science & Engineering						
2. Course Name	Advanced Web Technology Lab	**L**	**T**		**P**	
3. Course Code		**0**	**0**		**2**	
4. Type of Course (use tick mark)		Core (✓)	**PE()**		**OE ()**	
5. Pre-requisite (if any)		**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 = 0		**Tutorials = 0**	**Practical = 12**			

8. Course Description

To help the students to understand the concept of HTML, CSS, Javascript and PHP

9. Learning objectives:

Upon successful completion of the course in this discipline the student will be able to develop a complete dynamic website with database as backend.

10. Course Outcomes (COs):

- Hand on practice on HTML and learned the need and basics of CSS and the concepts of client side JavaScript
- How to design and develop a dynamic website.
- Import multimedia pages over the web.

11. List of Experiments

List of Experiments:

1. Create a Web Page using basic tags in html 5
2. Write a program to create all types of list in HTML
3. Create a table using Html 5 and CSS
4. Write a program using labels, radio buttons, and submit buttons
5. Create a simple webpage using HTML
6. Use frames to Include Images and Videos.
7. Add a Cascading Style sheet for designing the web page.
8. Design a web page with validation using JavaScript.
9. How to make all fields of a form mandatory in javascript
10. Create a registration form and validate it using javascript
11. Perform database connectivity in PHP

12	Create a dynamic web page using PHP

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

The students will be encouraged to learn using Virtual Link.

https://html-iitd.vlabs.ac.in/List%20of%20experiments.html

Semester V
MCA

<table>
<tr><td colspan="8">14. Name of the Department- Computer Science & Engineering</td></tr>
<tr><td>15. Course Name</td><td colspan="2">Software Project Management</td><td colspan="2">L</td><td colspan="2">T</td><td>P</td></tr>
<tr><td>16. Course Code</td><td colspan="2"></td><td colspan="2">3</td><td colspan="2">0</td><td>0</td></tr>
<tr><td colspan="3">17. Type of Course (use tick mark)</td><td colspan="2">Core (✓)</td><td colspan="2">PE()</td><td>OE ()</td></tr>
<tr><td>18. Pre-requisite (if any)</td><td colspan="2">Programming Lang. and Software Engg.</td><td>19. Frequency (use tick marks)</td><td>Even ()</td><td>Odd (✓)</td><td>Either Sem ()</td><td>Every Sem ()</td></tr>
<tr><td colspan="8">20. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</td></tr>
<tr><td colspan="3">Lectures = 36</td><td colspan="3">Tutorials = 0</td><td colspan="2">Practical = 0</td></tr>
<tr><td colspan="8">21. Course Description</td></tr>
<tr><td colspan="8">This course describes the key aspects of a software project. It begins with the job description of a software manager and then addresses those topics germane to successful software development management, including organizing the software development team; interfacing with other engineering organizations, assessing development standards; selecting the best approach and tailoring the process model; estimating software cost and schedule; planning and documenting the plan; staffing the effort; managing software cost and schedule during development; risk engineering; and continuous process improvement.</td></tr>
<tr><td colspan="8">22. Learning objectives:
To understand the methods used to evaluate and select projects for investment of funds
To gain knowledge on the principles and techniques of software project management
To introduce organization behavior and general management techniques used for project management</td></tr>
<tr><td colspan="8">23. Course Outcomes (COs):</td></tr>
<tr><td colspan="8">Apply project management concepts and techniques to an IT project.</td></tr>
<tr><td colspan="8">Identify issues that could lead to IT project success or failure.</td></tr>
<tr><td colspan="8">Explain project management in terms of the software development process.</td></tr>
<tr><td colspan="8">Describe the responsibilities of IT project managers</td></tr>
<tr><td colspan="8">24. Unit wise detailed content</td></tr>
<tr><td>Unit-1</td><td colspan="2">Number of lectures = 9</td><td colspan="5">PROJECT CONCEPTS AND ITS MANAGEMENT</td></tr>
<tr><td colspan="8">Software Project Categorization, Software VS other projects, Stakeholders, Project Success and Failure, Software project Activities Project life cycle models-ISO 9001 model-Capability Maturity Model-Project Planning-Project tracking-Project closure. Evolution of Software Economics – Software Management Process Framework: Phases, Software Management Planning / Project Organization and Responsibilities</td></tr>
<tr><td>Unit – 2</td><td colspan="2">Number of lectures = 8</td><td colspan="5">COST ESTIMATION</td></tr>
<tr><td colspan="8">Problems in Software Estimation – Algorithmic Cost Estimation Process, Function Points, SLIM (Software Life cycle Management), COCOMO II (Constructive Cost Model) – Estimating Web Application Development – Concepts of Finance, Activity Based Costing and Economic Value Added</td></tr>
</table>

(EVA) – Balanced Score Card.

Unit – 3	**Number of lectures = 10**	**SOFTWARE QUALITY MANAGEMENT**

Software Quality Factors – Software Quality Components – Software Quality Plan – Software Quality Metrics – Software Quality Costs – Software Quality Assurance Standard – Certification – Assessment. Software Configuration Management – Risk Management: Risk Assessment: Identification / Analysis / Prioritization

Risk Control: Planning / Resolution / Monitoring

Failure Mode and Effects Analysis (FMEA) ,Defect Management ,Cost Management.

Unit – 4	**Number of lectures = 19**	**PROJECT EVALUATION AND EMERGING TRENDS**

Strategic Assessment–Technical Assessment–Cost Benefit Analysis–Cash Flow Forecasting–Cost Benefit Evaluation Technique–Risk Evaluation–Software Effort Estimation. Emerging Trends: Import of the internet on project Management – people Focused Process Models

25. 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/

26. Books Recommended

Text Books

- Bob hughes and Mike Cotterell, "Software Project Management" second edition,1999.
- Royce, W. "Software Project Management: A Unified Framework", AddisonWesley, 1998.

Reference Books

- Ramesh Gopalaswamy , "Managing and global Software Projects", Tata McGraw Hill Tenth Reprint, 2011.
- Roger S.Pressman, "Software Engineering- A Practitioner's Approach", 7th Edition ,McGraw Hill, 2010.
- Daniel Galin, "Software Quality Assurance: from Theory to Implementation", Addison-Wesley, 2003.
- Fenton, N.E., and Pfleeger, S.L.. "Software Metrics: A Rigorous and Practical Approach, Revised" Brooks Cole, 1998.
- Demarco, T. and Lister, T. "Peopleware: Productive Projects and Teams, 2nd Ed.", Dorset House,1999.

Semester V
MCA

<table>
<tr><td colspan="8">Name of the Department- Computer Science & Engineering</td></tr>
<tr><td colspan="2">1. Course Name</td><td>Software Testing</td><td colspan="2">L</td><td colspan="2">T</td><td>P</td></tr>
<tr><td colspan="2">2. Course Code</td><td></td><td colspan="2">3</td><td colspan="2">0</td><td>0</td></tr>
<tr><td colspan="3">3. Type of Course (use tick mark)</td><td colspan="2">Core (✓)</td><td colspan="2">PE()</td><td>OE ()</td></tr>
<tr><td colspan="2">4. Pre-requisite (if any)</td><td></td><td>5. Frequency (use tick marks)</td><td>Even ()</td><td>Odd (✓)</td><td>Either Sem ()</td><td>Every Sem ()</td></tr>
<tr><td colspan="8">6. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)</td></tr>
<tr><td colspan="3">Lectures = 42</td><td colspan="2">Tutorials = 0</td><td colspan="3">Practical = 0</td></tr>
<tr><td colspan="8">7. Course Description</td></tr>
<tr><td colspan="8">This course presents the concepts and techniques for testing software and assuring its quality. Topics cover software testing at the unit, module, subsystem, and system levels, automatic and manual techniques for generating and validating test data, the testing process</td></tr>
<tr><td colspan="8">8. Learning objectives:
Appreciate the fundamentals of software testing and its application through the software life cycle.
Develop skills in designing and executing software tests suitable for different stages in the software life cycle.
Understand and appreciate the role of software testing in systems development, deployment and maintenance.
Develop a continuing interest in software testing, and obtain satisfaction from its study and practice.
Appreciate the responsibilities of software testers within software projects, the profession and the wider community.</td></tr>
<tr><td colspan="8">9. Course Outcomes (COs):

Students who have completed this course would have learned

• Various test procesess and continuous quality improvement
• Types of errors and fault models
• Methods of test generation from requirements
• Behavior modeling using UML: Finite state machines (FSM)
• Test generation from FSM models
• Input space modeling using combinatorial designs
• Combinatorial test generation
• Test adequacy assessment using: control flow, data flow, and program mutations
• The use of various test tools
• Application of software testing techniques in commercial environments</td></tr>
<tr><td colspan="8">10. Unit wise detailed content</td></tr>
<tr><td colspan="2">Unit-1</td><td>Number of lectures = 09</td><td colspan="5"></td></tr>
</table>

Introduction: 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.
Verification Testing: Verification Methods, SRS Verification, Software Design Document Verification, Code Reviews, User Documentation Verification, Software Project Audits.

Unit – 2	**Number of lectures = 09**	

Functional Testing: Boundary Value Analysis, Equivalence Class Testing, Decision Table Based Testing, Cause Effect Graphing Technique.
Structural Testing: Identification of Independent Paths: Control Flow Graph, DD-Paths, Cyclomatic Complexity, Graph Matrix, Control Flow Testing, Data Flow Testing, Slice Based Testing, Mutation testing.

Unit – 3	**Number of lectures = 09**	

Use Case Testing: 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,
Prioritization of test cases for Regression Testing: Regression Testing, Regression Test Case Selection, Prioritization guidelines, Priority category Scheme, Code Coverage Techniques for Prioritization of Test Cases, Risk Analysis.

Unit – 4	**Number of lectures = 09**	

Testing Activities: 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.
Metrics and Models in Software Testing: What are Software Metrics, categories of Metrics, object Oriented Metrics used in testing, What should we measure during testing?, Software Quality Attributes.
Prediction Model: Reliability Modes, Fault Prediction Model.

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.
https://elearning.sgtuniversity.ac.in/course-category/Software Testing

12. Books Recommended

Text Books

- William Perry, “Effective Methods for Software Testing”, John Wiley & Sons, New York, Van Nostrand Reinhold, New York, 2nd Ed., 1995.
- Cem Kaner, Jack Falk, Nguyen Quoc, “Testing Computer Software”, Van Nostrand Reinhold, New York, 2nd Ed., 1993.
- Boris Beizer, “Software Testing Techniques”, Second Volume, Van Nostrand Reinhold, New York, . 2nd Ed., 1990.

Reference Books

Paul C. Jorgenson, Software Testing A Craftsman‟s approach, CRC Press, 1997.

Roger S. Pressman, “Software Engineering – A Practitioner‟s Approach”, McGraw-Hill International Edition, New Delhi, 5th Ed., 2001.

Boris Beizer, “Black-Box Testing – Techniques for Functional Testing of Software and

Systems”, John Wiley & Sons Inc., New York, 1995.

Program Electives
MCA

1. Name of the Department- Computer Science & Engineering						
2. Course Name	**System Network Administration**	**L**	**T**		**P**	
3. Course Code		**3**	**0**		**0**	
4. Type of Course (use tick mark)		**Core ()**	**PE(✓)**		**OE ()**	
5. Pre-requisite (if any)		**Frequency (use tick marks)**	Even ()	Odd ()	Either Sem ()	Every Sem ()
6. Total Number of Lectures, Tutorials, Practical (assuming 12 weeks of one semester)						
Lectures = 36		**Tutorials = 0**	**Practical = 0**			
7. Course Description						
This course aims to give students in depth information about security, host administration and Unix commands						
8. Learning objectives: To learn essential systems administration skills related to operating systems, system and network						

service administration, computer and information security and directory services administration

9. Course Outcomes (COs):

Upon the completion of this course, the student will be able to:

- To Install the Unix operating system, and apply operating system updates, and configuration changes.
- To Install and configure new hardware/software
- To Manage user accounts
- To Perform backups of data
- To Assess system security

10. Unit wise detailed content

Unit-1	**Number of lectures = 9**	

Network Administration:-system administrator, network administrator, phases of network administration, addresses in TCP/IP model, IP addressing, Sub netting, Supernetting, NAT,Basic Concepts of proxy server, web server, DNS and their respective configuration settings. Various Interconnecting Devices; Hub, Switch, Bridges, Routers, Gateway, repeater, brouter. Commands used in troubleshooting of TCP/IP: ping, netstat, tracert, traceroute, ifconfig and route command.

Unit – 2	**Number of lectures = 09**	

System Administration: Introduction to system Administration, goals of system administrator, role of network and system administrator, unix operating system, comparison of various operating systems, file system-NFS, UFS and NTFS, System performance tuning

Unit – 3	**Number of lectures = 9**	

Host and Network Security: Types of computer security, aspects of security, types of attacks, network security mechanisms, authentication and authorization for remote access, access control and monitoring, Access Control Models ABAC,DAC,MAC,RBAC, firewall, filtering rules, detection and prevention of denial of service attack

Unit – 4	**Number of lectures = 09**	**Design of experiments & Time series and forecasting**

Host management:-installation of Unix, Linux, windows OS, booting process in various OS, File allocation methods, User accounts, controlling user resources, Unix Commands, advantages and disadvantages of Shell scripting.

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

12. Books Recommended

Text Books

1. The unix programming environment, Brain Kemighen & Rob Pike, Pearson Education India; 1 edition, 2015.

Reference Books

1. Design of the Unix operating system, AT&T Bell Labs Maurice J. Bach, Pearson Education India; 1 edition, 2015.
2. Advanced Unix programmer's Guide, Stephen Prato Bpb publisher, 2008.
3. Unix Concepts and applications-Featuring SCO Unix and Linux, Sumitabha Das

1. Name of the Department- Computer Science & Engineering						
2. Course Name	**Advanced Database management System**	**L**	**T**		**P**	
3. Course Code		**3**	**0**		**0**	
4. Type of Course (use tick mark)		Core (✓)	**PE()**		**OE ()**	
5. Prerequisite (if any)	DBMS	**6. Frequency (use tick marks)**	Even ()	Odd (✓)	Either Sem ()	Every Sem ()
7. Total Number of Lectures, Tutorials, Practical (assuming 12weeks of one semester)						
Lectures = 36		**Tutorials = 0**	**Practical = 0**			
8. Course Description						
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.

9. Learning objectives:

- To understand the basic concepts and terminology related to DBMS and Relational Database Design
- To the design and implement Distributed Databases.
- To understand advanced DBMS techniques to construct tables and write effective queries, forms, and reports

10. Course Outcomes (COs):

- Exposure for students to write complex queries including full outer joins, self-join, sub queries, and set theoretic queries.
- Know how of the file organization, Query Optimization, Transaction management, and database administration techniques

11. Unit wise detailed content

Unit-1	**Number of lectures = 9**	

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

Unit – 2	**Number of lectures = 9**	

Processing of joins, materialized vs. pipelined processing, query transformation rules, DB transactions, ACID properties, interleaved executions, schedules, serializability

Unit – 3	**Number of lectures = 9**	

Correctness of interleaved execution, Locking and management of locks, 2PL, deadlocks, multiple level granularity, CC on B+ trees, Optimistic CC

Unit – 4	**Number of lectures = 9**	

Time stamped, lock based techniques, Multiversion approaches, Comparison of CC methods, dynamic databases, Failure classification, recovery algorithm, XML and relational databases

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

- 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.

1. Name of the Department- Computer Science & Engineering						
2. Course Name	**Programming in Python**	**L**	**T**		**P**	
3. Course Code		**3**	**0**		**0**	
4. Type of Course (use tick mark)		**Core ()**	**PE(✓)**		**OE ()**	
5. Pre-requisite (if any)	Basics of Programming	**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

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.

9. Learning objectives:

- To Learn concepts of various Python script at the shell prompt, Python types, expressions to solve relative problems

10. Course Outcomes (COs):

- To utilize high-level data types such as lists and dictionaries
- To import and utilize a module read from and write to a text file
- Understand the difference between mutable and immutable types
- To demonstration of IDE‟s: IDLE, IPython, IPython Notebook, hosted environments

11. Unit wise detailed content

Unit-1	**Number of lectures = 09**	

Introduction, Python basic data types, files, functions, and error handling. Working with Data, introduction to tuples, lists, dictionaries, and sets. Students will also learn how to effectively use Python's very powerful list processing primitives such as list comprehensions. Finally, this section covers critical aspects of Python's underlying object model including variables, reference counting, copying, and type checking.

Unit – 2	**Number of lectures = 09**	

More information about how to organize larger programs into functions. A major focus of this section is on how to design functions that are reliable and can be easily reused in other settings. Also covers technical details of functions including scoping rules and documentation strings. Modules and Libraries. How to organize programs into modules and details on using modules as a tool for creating extensible programs. Concludes with a tour of some of the most commonly used library modules including those related to system administration, text processing, subprocesses, XML parsing, binary data handling, and databases. Also includes information on how to install third-party library modules

Unit – 3	**Number of lectures = 09**	

An introduction to object-oriented programming in Python. Describes how to create new objects, overload operators, and utilize Python special methods. Also covers basic principles of object oriented programming including inheritance and composition. Inside the Python Object System. A detailed look at how objects are implemented in Python. Major topics include object representation, attribute binding, inheritance, memory management, and special properties of classes including properties, slots, and private attributes.

Unit – 4	**Number of lectures = 9**	

This includes effective use of documentation strings, program testing using both the doctest and unittest modules, and effective use of assertions. The Python debugger and profiler are also described. Iterators and Generators. Covers the iteration protocol, iterable objects, generators and generator expressions. A major focus of this section concerns the use of generators to set up data processing pipelines--a particularly effective technique for addressing a wide variety of common systems programming problems (e.g., processing large datafiles, handling infinite data streams, etc.). Text I/O Handling. More information on text-based I/O. Topics include text generation, template strings, and Unicode. Some Advanced Topics. A variety of more advanced programming topics

including variable argument functions, anonymous functions (lambda), closures, decorators, static and class methods, and packages. Python Integration Primer. A survey of how Python is able to interact with programs written in other programming languages. Topics include network programming, accessing C code, COM extensions, Python, and Iron Python.

12. 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.

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

13. Books Recommended

Text Books

- Learning to Program Using Python by Cody Jackson
- Python for complete beginners by Dr. Martin Jones

Reference Books

- 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

1. Name of the Department: CSE						
2. Course Name	**Compiler design**	**L**	**T**		**P**	
3. Course Code		**3**	**0**		**0**	
4. Type of Course (use tick mark)		**Core ()**	**PE(√)**		**OE()**	
5. Pre-requisite (if any)	TOC	**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. Brief Syllabus

COURSE CONTENT

Unit I: Introduction

Introduction to Compiler, Phases and passes, Bootstrapping, Finite state machines and regular expressions and their applications to lexical analysis, Optimization of DFA, implementation of lexical analyzers, lexical-analyzer generator, LEX-compiler, Ambiguity. The syntactic specification of programming languages: Context free grammars, derivation and parse trees, capabilities of CFG.

Unit II: Basic Parsing Techniques

Parsers, Shift reduce parsing, operator precedence parsing, top down parsing, predictive parsers Automatic Construction of efficient Parsers: LR parsers, the canonical Collection of LR (0) items, constructing SLR parsing tables, constructing

Canonical LR parsing tables, Constructing LALR parsing tables, using ambiguous grammars, an automatic parser generator, and implementation of LR parsing tables.

Unit III: Syntax-directed Translation

Syntax-directed Translation schemes, Implementation of Syntax directed Translators, Intermediate code, postfix notation, Parse trees & syntax trees, three address code, quadruple & triples, Boolean expressions, statements that alter the flow of control, postfix translation, translation with a top down parser. More about translation: Array references in arithmetic expressions, procedures call.

Unit IV: Symbol Tables&Code Generation

Data structure. for symbols tables, representing scope information. Run-Time Administration: Implementation of simple stack al-location scheme, storage allocation in block structured language. Error Detection & Recovery: Lexical Phase errors, syntactic phase errors semantic errors.

9. Learning objectives:

- Provide an understanding of the fundamental principles in compiler design
- Provide the skills needed for building compilers for various situations that one may encounter in a career in Computer Science.
- Learn the process of translating a modern high-level language to executable code required for compiler construction.

10. Course Outcomes:

At the end of the course student will be able to:

- Understand fundamentals of compiler and identify the relationships among different phases of the compiler.
- Understand the application of finite state machines, recursive descent, production rules, parsing, and language semantics.
- Analyze & implement required module, which may include front-end, back-end, and a small set of middle-end optimizations.
- Use modern tools and technologies for designing new compiler.

11. Unit wise detailed content

Unit-1	**Number of lectures = 10**	**Title of the unit: Introduction**

Introduction to Compiler, Phases and passes, Bootstrapping, Finite state machines and regular expressions and their applications to lexical analysis, Optimization of DFA, implementation of lexical analyzers, lexical-analyzer generator, LEX-compiler, Ambiguity. The syntactic specification of programming languages: Context free grammars, derivation and parse trees, capabilities of CFG.

Unit – 2	**Number of lectures =8**	**Title of the unit: Basic Parsing Techniques**

Parsers, Shift reduce parsing, operator precedence parsing, top down parsing, predictive parsers Automatic Construction of efficient Parsers: LR parsers, the canonical Collection of LR (0) items, constructing SLR parsing tables, constructing Canonical LR parsing tables, Constructing LALR parsing tables, using ambiguous grammars, an automatic parser generator, and implementation of LR parsing tables.

Unit – 3	**Number of lectures = 8**	**Title of the unit: Syntax-directed Translation**

Syntax-directed Translation schemes, Implementation of Syntax directed Translators, Intermediate code, postfix notation, Parse trees & syntax trees, three address code, quadruple & triples, Boolean expressions, statements that alter the flow of control, postfix translation, translation with a top down parser. More about translation: Array references in arithmetic expressions, procedures call.

Unit – 4	**Number of lectures = 10**	**Title of the unit: Symbol Tables&Code Generation**

Data structure. for symbols tables, representing scope information. Run-Time Administration: Implementation of simple

stack al-location scheme, storage allocation in block structured language. Error Detection & Recovery: Lexical Phase errors, syntactic phase errors semantic errors.
12. Brief Description of self learning / E-learning component
13. Books Recommended (2 Text Books + 2 Reference Books)
• ALFRED V AUTOR AHO, JEFFREY D AUTOR ULLMAN "Principles of Compiler Design".
• V Raghvan, " Principles of Compiler Design", TMH
REFERENCE BOOKS
• Aho, Sethi & Ullman, "Compilers: Principles, Techniques and Tools", Pearson Education2
• Charles Fischer and Ricard LeBlanc," Crafting a Compiler with C", Pearson Education

1. Name of the Department: CSE						
2. Course Name	**Distributed Operating system**	**L**	**T**		**P**	
3. Course Code	**13020610**	**3**	**0**		**2**	
4. Type of Course (use tick mark)		**Core ()**	**PE(√)**		**OE()**	
5. Pre-requisite (if any)	DWDM,O.S	**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. Brief Syllabus

COURSE CONTENT

Unit I: Introduction

Definition - Evolution- Goals of distributed systems, system models- Issues in the design of distributed systems- Distributed com-puting environment

Unit II: COMMUNICATION

Message Passing – Features and Issues -Synchronization-Buffering - Process Addressing - Failure Handling - Remote procedure call (RPC): Model – Implementation - Stub generation - RPC messages – Marshaling - server Management - Call semantics - communication protocols for RPC-Client server binding – RMI.

Unit III : DISTRIBUTED SHARED MEMORY
Distributed shared memory- Design and implementation issues- Sequential consistency - Release consistency, Process migration Features & Mechanism

Unit IV:SYNCHRONIZATION
Synchronizing physical clocks - Logical clocks - Distributed coordination – Event Ordering – Mutual Exclusion – Deadlock - Elec-tion algorithms

DISTRIBUTED FILE SYSTEMS
Introduction – File Models – File accessing, sharing and caching - File Replication – Atomic transactions Case Study HADOOP. : Resource and process management - Task assignment approach - Load balancing approach - Load sharing approach

9. Learning objectives:

1. Familiarize the students with the basics of distributed computing systems.
2. To introduce the concepts of distributed file systems, shared memory and message passing systems, synchronization and resource management.

10. Course Outcomes:

1. Verify and analyze the time complexity of the algorithms related to distributed computing.
2. Design and develop various algorithms for problems in distributed computing
3. Compare various resource allocation stratagies.

At the end of the course student will be able to:

11. Unit wise detailed content

Unit-1	**Number of lectures = 10**	**Title of the unit: Introduction**

Definition - Evolution- Goals of distributed systems, system models- Issues in the design of distributed systems- Distributed com-puting environment

Unit - 2	**Number of lectures =8**	**Title of the unit: COMMUNICATION**

Message Passing – Features and Issues -Synchronization-Buffering - Process Addressing - Failure Handling - Remote procedure call (RPC): Model – Implementation - Stub generation - RPC messages – Marshaling - server Management - Call semantics - communication protocols for RPC-Client server binding – RMI.

Unit - 3	**Number of lectures = 8**	**Title of the unit: DISTRIBUTED SHARED MEMORY**

Distributed shared memory- Design and implementation issues- Sequential consistency - Release consistency, Process migration Features & Mechanism

Unit - 4	**Number of lectures = 10**	**Title of the unit: SYNCHRONIZATION**

Synchronizing physical clocks - Logical clocks - Distributed coordination – Event Ordering – Mutual Exclusion – Deadlock - Elec-tion algorithms.

Introduction – File Models – File accessing, sharing and caching - File Replication – Atomic transactions Case Study HADOOP. : Resource and process management - Task assignment approach - Load balancing approach - Load sharing approach

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

Online Video Lectures of DOS
13. Books Recommended (1Text Books + 4 Reference Books)
1. George Colouris, Jean Dollimore and Tim Kindberg, "Distributed Systems – Concepts and Design", Pearson Education Private Limited, New Delhi, 2001
REFERENCE BOOKS
1. Gerard Tel, "Introduction to Distributed algorithms", Cambridge University Press, USA, 2000.
2. Andrzej Goscinski, "Distributed Operating Systems, the logical Design", Addison Wesley Publishing Company, USA, 1991.
3. Tanenbaum, "Modern Operating Systems", Prentice Hall of India, New Delhi, 1999.

1. Name of the Department- Computer Science and Engineering						
2. Course Name	**Ethical Hacking**	**L**	**T**		**P**	
3. Course Code		**3**	**0**		**0**	
4. Type of Course (use tick mark)		**Core ()**	**PE(✔)**		**OE ()**	
5. Pre-requisite (if any)	**Knowledge of cyber security**	**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 goal of this course is to help you master an ethical hacking methodology that can be used in a penetration testing or ethical hacking situation. You walk out the door with ethical hacking skills that are highly in demand.						
9. Learning objectives: The objective of this course is to give knowledge of computer systems architecture,						

programming, operating systems and databases, alongside specialist topics in digital forensics, legal issues, networking, ethical hacking and computer security.

10. Course Outcomes (COs):

On completion of this course, the students will be able to

- Establish type of attack on a given system.
- Analyze nature and type of attack.
- Simulate different types of attacks using tools
- Design a secure system for protection from the various attacks by determining the need of security from various departments of an organization.

11. Unit wise detailed content

Unit-1	**Number of lectures = 09**	**Title of the unit: Introduction to Ethical Hacking**

Introduction, Networking & Basics, Foot Printing, Google Hacking, Scanning, Windows Hacking, Linux Hacking, Trojans &
Backdoors, Virus & Worms, Proxy & Packet Filtering, Denial of Service, Sniffer, Social Engineering,

Unit – 2	**Number of lectures = 08**	**Title of the unit: Introduction to Networking**

Introduction to Computer Systems and Networks , information systems and networks (including wireless networks) and their role in industry business and society, System and Network Vulnerability and Threats to Security , various types of attack and the various types of attackers in the context of the vulnerabilities associated with computer and information systems and networks Physical Security, Steganography, Cryptography, Wireless Hacking, Firewall & Honeypots, IDS & IPS, Vulnerability, Penetration Testing, Session Hijacking.

Unit – 3	**Number of lectures = 08**	**Title of the unit: Fundamentals of Hacking**

Hacking Web Servers, SQL Injection, Cross Site Scripting, Exploit Writing, Buffer Overflow, Reverse Engineering, Email Hacking, Incident Handling & Response, Bluetooth Hacking, Mobiles Phone Hacking. – Intrusion detection system – NIDS, HIDS – Penetrating testing process – Web Services – Reducing transaction risks.

Unit – 4	**Number of lectures = 10**	**Title of the unit: Ethical issues**

An introduction to the particular legal, professional and ethical issues likely to face the domain of ethical hacking. Ethical responsibilities, professional integrity and making appropriate use of the tools and techniques associated with ethical hacking.

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/
Journal papers; Patents in the respective field.

13. Books Recommended

- Hands-On Ethical Hacking and Network Defense – By Michael T. Simpson, Kent Backman,

James Corley
• Official Certified Ethical Hacker Review Guide – By Steven DeFino, Barry Kaufman, Nick Valenteen
• The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy (Syngress Basics Series) [Paperback]

1. Name of the Department: CSE						
2. Course Name	Microprocessor and Interfacing	**L-3**	**T-0**		**P-0**	
3. Course Code	13020673					
4. Type of Course (use tick mark)		**Core ()**	**PE**(√)		**OE()**	
5. Pre-requisite (if any)		**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**			
Brief Syllabus: • The 8085 processor						

- The 8086 microprocessor
- instruction set of 8086
- Interfacing device

8. Learning objectives:

The student will be able to select an appropriate 'architecture' or program design to apply to a particular situation; e.g. an interrupt-driven I/O handler for a responsive real-time machine. Following on from this, the student will be able to design and build the necessary programs.

9. Course Outcomes:

i. Recall and apply a basic concept of digital fundamentals to Microprocessor based personal computer system.

ii. Identify a detailed s/w & h/w structure of the Microprocessor.

iii. Illustrate how the different peripherals (8255, 8253 etc.) are interfaced with Microprocessor.

10. Unit wise detailed content

Unit-1	**Number of lectures =9**	**Title of the unit: THE 8085 PROCESSOR**

THE 8085 PROCESSOR : Introduction to microprocessor, Pin layout,Architecture, instruction set, interrupt structure,Addressing modes and Assembly language programming for arithmetic functions.

Unit – 2	**Number of lectures = 9**	**Title of the unit: THE 8086 MICROPROCESSORARCHITECTURE**

THE 8086 MICROPROCESSOR ARCHITECTURE : Architecture, block diagram of 8086, Pin layout, memory segmentation and physical address computations,Interrupt logic description of various signals.

Unit – 3	**Number of lectures =9**	**Title of the unit: INSTRUCTION SET OF 8086**

INSTRUCTION SET OF 8086: Instruction execution timing, assembler instruction format, addressing modes, Instructions set ,directives , programming examples for mathematical function.

Unit – 4	**Number of lectures = 9**	**Title of the unit:INTERFACING DEVICE**

INTERFACING DEVICE: Additional IC Operations: 8255, 8254, 8259A, 8237.
Interfacing of LED's, LCD, keyboard, Seven segment motor.

11. 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.

12. Books Recommended (1 Text Books + 3-4 Reference Books)

- Microprocessor Architecture, Programming & Applications with 8085 : Ramesh S Gaonkar; Wiley Eastern Ltd.
- The Intel Microprocessors 8086- Pentium processor : Brey; PHI
- Microprocessors and interfacing : Hall; TMH
- The 8088 & 8086 Microprocessors-Programming, interfacing,Hardware & Applications :Triebel & Singh; PHI
- Microcomputer systems: the 8086/8088 Family: architecture, Programming & Design : Yu-Chang Liu & Glenn A Gibson; PHI
- Advanced Microprocessors and Interfacing : Badri Ram; TMH