

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SYLLABUS COVERAGE

FOR B.TECH III YEAR - II SEM(R17)

(2019-20)



MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous Institution – UGC, Govt. of India)

**(Affiliated to JNTUH, Hyderabad, Approved by AICTE - Accredited by NBA & NAAC – ‘A’ Grade -
ISO 9001:2015 Certified)**

**Maisammaguda, Dhulapally (Post Via. Hakimpet), Secunderabad – 500100,
Telangana State, INDIA.**

Sno	Subject Code	Subject Name
1	R17A0522	DISTRIBUTED SYSTEMS
2	R17A0518	OBJECT ORIENTED ANALYSIS AND DESIGN
3	R17A0464	EMBEDDED SYSTEMS
4	R17A0520	SOFTWARE TESTING METHODOLOGIES
5	R17A0554	PYTHON PROGRAMMING
6	R17A0519	WEB TECHNOLOGIES

III B.Tech II Semester

Subject: Distributed Systems

Branch: CSE

Subject Code: R17A0522

Academic Year: 2019-2020

TEXTBOOKS:

1. Distributed Systems, Concepts and Design, George Coulouris, J Dollimore and Tim Kindberg, Pearson Education, 4th Edition, 2009.

S.NO	UNIT	Topics	Reference
1	I	Introduction	T1: 1-33
2		Examples of Distributed systems	T1: 1-33
3		Resource sharing and web	T1: 1-33
4		challenges	T1: 1-33
5		System models: Introduction	T1:37-76
6		Architectural and Fundamental models	T1:37-76
7		networking and Internetworking	T1:37-76
8		Interposes Communication	T1:37-76
9	II	Time and Global States: Introduction	T1:595-626
10		Clocks, events and Process states	T1:595-626
11		Synchronizing physical clocks	T1:595-626
12		logical time and logical clocks	T1:595-626
13		global States	T1:595-626
14		distributed debugging	T1:595-626
15		Coordination and Agreement: Introduction	T1: 629-671
16		Distributed mutual exclusion	T1: 629-671
17		Elections	T1: 629-671
18		Multicast communication	T1: 629-671
19		consensus and related problems	T1: 629-671
20	III	Inter process Communication: Introduction	T1:145-181

21		The API for the Internet Protocols	T1:145-181
22		External Data Representation and Marshalling	T1:145-181
23		Client –Server Communication	T1:145-181
24		Group Communication	T1:145-181
25		Case Study: IPC in UNIX	T1:145-181
26		Distributed Objects and Remote Invocation: Introduction	T1:185-225
27		Communication between distributed objects	T1:185-225
28		Remote Procedure Call	T1:185-225
29		Events and Notifications	T1:185-225
30		Case Study: JAVA RMI	T1:185-225
31	IV	Distributed File Systems: Introduction	T1:521-563
32		File Service Architecture	T1:521-563
33		Case Study 1: Sun Network File System	T1:521-563
34		Case Study2:The Andrew File System	T1:521-563
38		Distributed Shared Memory: Introduction	T1:262-274
39		Design and Implementation issues	T1:262-274
40		Sequential consistency and Ivy case study	T1:262-274
41		Release consistency and Munin case study	T1:262-274
42		Other consistency models	T1:262-274
43	V	Transactions and Concurrency control: Introduction	T1:675-720
44		Transactions	T1:675-720
45		Nested Transactions	T1:675-720
46		Locks	T1:675-720
47		optimistic concurrency control	T1:675-720
48		Timestamp ordering	T1:675-720
49		Comparison of methods for	T1:675-720

		concurrency control	
50		Distributed Transactions: Distributed Transactions: Introduction	T1:727-761
51		Flat and Nested Distributed Transactions	T1:727-761
52		Atomic commit protocols	T1:727-761
53		Concurrency control in distributed transactions	T1:727-761
54		Distributed deadlocks	T1:727-761
55		Transaction recovery	T1:727-761

OBJECT ORIENTED ANALYSIS AND DESIGN

Subject Code: R17A0518

AcademicYear:2019-2020

TEXTBOOKS:

1. GRADY Booch, James Rumbaugh, Ivar Jacobson: The Unified Modeling Language User Guide, Pearson Education
2. Hans-Erik Erikson, Magnus Penker, Brian Lyons, David Fado UML 2 toolkit, WILEY- Dreamtech India Pvt Ltd

REFERENCES:

1. Meilir Page Jones: Fundamentals of Object Oriented Design in UML, Pearson Education
2. Pascal Roques: Modeling Software Systems Using UML2, WILEY, Dreamtech India Pvt.
3. Atul Kahate: Object Oriented Analysis & Design, The McGraw Hill Companies
4. Craig Larman Applying UML and Patterns: An introduction to Object Oriented Analysis and Design and Unified Process, Pearson Education
5. Mark Priestley: Practical Object-oriented Design with UML, TATA McGraw Hill

SYLLABUS COVERAGE

units	Topics	Course Learning Outcomes	Reference
UNIT-1	Importance of Modeling ,Principles of Modeling ,Object Oriented Modeling	Explain the importance of modeling	T1:26-32
	Conceptual model of UML. Architecture Software Development Life Cycle	Demonstrate the Conceptual model of UML and SDLC.	T1:39-57
	Classes, Modeling the vocabulary, Modeling the distribution of responsibilities modeling non structure things, modeling primitive types of systems	Illustrate classes and relationships mechanisms and diagram	T1:69-95
	Relationships: Terms and Concepts. modeling single dependency, modeling single inheritance, modeling structural relationship, creating webs of relationships	Illustrate relationships mechanisms and diagrams. Demonstrate common	T1:97-155
	Common mechanisms and diagrams Notes, Stereotypes, Tagged values and constraints modeling comments, new building blocks new properties semantics extending UML	mechanisms and diagrams	T1:157-201
	Diagrams, view and models, modeling Diagrams, view and models, modeling different level of abstraction, modeling complex views	Explain complex views of system	T1:127-213
	Advanced structural modeling, advanced classes classifiers, modeling semantics of classes choosing right kind of classifier	Define Advanced structural modeling Techniques	T1:214-223
Unit-2	Advanced relationships advanced dependencies, generalization, association, realization Refinement relationships, creating webs of relationships	Explain advanced relationships	T1:229-239
	Interfaces, Types, Roles, Realization Modeling seams in a seam. Static and dynamic types, Making interfaces understandable & approachable	Illustrate Interfaces and their modeling techniques	T1:267-278
	Common modeling techniques ,simple collaboration forward and reverse engineering	List interaction Modeling techniques	T1:244-261

	Instances terms and concepts common modeling techniques modeling concrete instances	Describe instances and their modeling techniques	T1:281-311
	Modeling Prototypical Instances. Object Modeling Prototypical Instances. Object Forward & Reverse engineering	State Prototypical modeling techniques	T1:281-311
Unit-3	Interaction terms and concepts common modeling techniques modeling flow of control	Demonstrate interaction diagram	T1:312-360
	Interaction diagrams terms and concepts common modeling techniques,flow control by time ordering,flow control by organization forward and reverse engg	Illustrate Interfaces and their modeling techniques	T1:355-378
	Use case terms and concepts common modeling techniques modeling behavior of the element	Demonstrate use case diagrams	T1:382-433
	Use case diagrams: Common Modeling Techniques: Modeling the Context of a system, Modeling the requirements of a system forward & reverse engg	Describe Use cases and their modeling techniques	T1:434-438
	Activity Diagrams: Terms and Concepts common modeling techniques modeling workflow modeling in operation forward reverse engg	State Activity diagrams and their modeling techniques	T1:279-295
	Events and signals terms and concepts common modeling techniques modeling family of signals modeling exceptions	Describe Events And Signals	T1:299-306
UNIT4	State Machines: Terms and Concepts common modeling techniques modeling lifetime of an object	Illustrate State machines and their modeling techniques	T1:312-328
	Processes and Threads: Terms and concepts, Modeling multiple Flows of Control, modeling inter process communication	List Process and threads modeling techniques	T1:333-342
	Components terms and concepts modeling executable and libraries modeling tables files and documents modeling an API	Describe Component and deployment and their modeling techniques	T1:367-377
	Deployment: Terms and Concepts, Modeling Processors and Devices	Demonstrate Deployment diagram	T1:382-389
	Component diagram terms and concepts modeling source code an executable release,physical database adaptive systems forward and reverse engg	Demonstrate Component and deployment diagram	T1:416-425
Unit-5	Deployment diagram terms and concepts modeling an embedded systems modeling client server system a fully distributed system forward and reverse engg	Demonstrate Deployment Diagram client and server system	T1:429-438
	Patterns and Framework Artifact Diagrams The unified library Application	Illustrate Patterns and Framework	T1: 403-411
	Artifact Diagrams The unified library Application	List Artifacts diagrams and library diagram	T1: 476-477

(R17A0518) OBJECT ORIENTED ANALYSIS AND DESIGN

Objectives:

- To Understand the Object Oriented Life Cycle
- To Know how to identify Objects, Relationships, Services and Attributes through UML
- To Understand the Usecase Diagram
- To Know the Object Oriented Design Process
- To Know about Software Quality and Usability

UNIT-I

Introduction to UML, Importance of Modeling, Principles of Modeling, Object oriented modeling, Conceptual model of the UML, Architecture of UML, Software Development Life Cycle.

UNIT-II

Basic Structural Modeling, Classes, Relationships, Common Mechanisms, Basic Diagrams, Advanced Structural Modeling, Advanced Classes, Advanced Relationships, Interfaces, Types and Roles, Packages. Class and Object Diagrams, Terms, Concepts, Modeling Techniques for Class Diagrams

UNIT-III

Basic Behavioral Modeling-I, Interactions, Interaction Diagrams.

Basic behavioral Modeling-II, Usecases, Use case Diagrams, Activity Diagrams.

UNIT-IV

Advanced Behavioral Modeling, Events and Signals, State Machines, Processes and Threads, Time and Space, State Chart Diagrams. Architectural Modeling, Component, Deployment, Component Diagrams, Deployment Diagram.

UNIT-V Case Study, The Unified Library application

TEXT BOOKS :

- 1.Grady Booch, James Rumbaugh, Ivar Jacobson : The Unified Modeling Language User Guide, Pearson Education.
- 2.Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado: UML 2 Toolkit, WILEY-Dreamtech India Pvt. Ltd

III B.Tech II Semester

Subject: EMBEDDED SYSTEMS

Subject Code: R17A0464

Branch: CSE

Academic Year: 2019-2020

TEXT BOOKS:

T1. Introduction to Embedded Systems - Shibu k v, Mc Graw Hill Education.

References:

T2. Advanced Microprocessors and Peripherals – A. K. Ray and K.M. Bhurchandi, TMH, 2nd Edition 2006

T3. Embedded Systems, Raj Kamal, Second Edition TMH.

SYLLABUS COVERAGE

Unit	Topics	Course Learning Outcomes	Page Reference
UNIT I	Introduction to Microprocessors and Microcontrollers: 8086 Microprocessor	Explain the importance of Microprocessors and Microcontrollers	T1:19-21
	Architecture of 8086, Register Organization, Memory Segmentation, Signal descriptions	Illustrate the features of 8086 Microprocessor in detail	T2:2-14
	8086 Addressing Modes and Instruction set	Define various addressing modes and instructions of 8086	T2:38-72
	8051 Microcontroller: Introduction, 8051 Architecture, I/O Ports, Memory Organization, Instruction set of 8051.	Illustrate the features of 8051 microcontroller in detail	T2:648-665
UNIT II	Introduction to Embedded Systems: Introduction, History of embedded systems Classification of embedded systems based on generation and complexity, Purpose of embedded systems,	Define Embedded systems and its importance	T1:3-8
	Applications of embedded systems, and characteristics of embedded systems, Operational and Non-operational attributes of embedded systems.	Explain the characteristics of embedded systems	T1:72-79

UNIT III	Typical Embedded System: Elements of ES, Core of ES, Sensors and Actuators	Explain the components of Embedded Systems	T1:15-23 T1:26-28 T1:35
	On-board communication interfaces-I2C, SPI, Parallel interface	Explain different On-board communication interfaces	T1:45-48 T1:50-51
	External communication interfaces-RS232, USB, infrared, Bluetooth, Wi-Fi, ZigBee, GPRS	Explain different external communication interfaces	T1:51-59
UNIT IV	Embedded Firmware Design and Development: Introduction, Embedded firmwaredesign approaches-super loop based approach, operating system based approach;	Explain Embedded firmware design approaches	T1:59 T1:303-306
	Embedded firmware development languages-assembly language based development, high level language based development	Explain Embedded firmware development	T1:306-315
UNIT V	Embedded Programming Concepts: Data types, Structures, Modifiers, Loops and Pointers, Macros and Functions	Define C Programming embedded Concepts	T1:320-325 T1:333-345 T1:349-356 T3:235-258
	object oriented Programming, Embedded Programming in C++ & JAVA	Explain Embedded programming	T3:262-269

Introduction D

EMBEDDED SYSTEMS



SHIBUKV

INTRODUCTION TO EMBEDDED SYSTEMS



HHW DELLt

New Delhi New York St Louis San Francisco Auckland Bogotá Caracas

Contents

<i>Preface</i>	<i>xiii</i>
----------------	-------------

<i>Acknowledgements</i>	<i>xvii</i>
-------------------------	-------------

Part I

Embedded System: Understanding the Basic Concepts

1. Introduction to Embedded Systems	3
--	----------

1.1 What is an Embedded System?	4
1.2 Embedded Systems vs. General Computing Systems	4
1.3 History of Embedded Systems	5
1.4 Classification of Embedded Systems	6
1.5 Major Application Areas of Embedded Systems	7
1.6 Purpose of Embedded Systems	8
1.7 'Smart' Running Shoes from Adidas—The Innovative Bonding of Lifestyle with Embedded Technology	11
Summary	13
Keywords	13
Objective Questions	14
Review Questions	14

2. The Typical Embedded System	15
---------------------------------------	-----------

2.1 Core of the Embedded System	17
2.2 Memory	28
2.3 Sensors and Actuators	35
2.4 Communication Interface	43
2.5 Embedded Firmware	50
2.6 Other System Components	60
2.7 PCB and Passive Components	64
Summary	64
Keywords	64

Keywords 221

€ eslp■ ail<I OuvelapzarnJ nf F.rabrx zJzd Prarlzie5

8. Embedded Hardware Design and Development	228
8.1 Analog Electronic Components	229
8.2 Digital Electronic Components	230
8.3 VLSI and Integrated Circuit Design	243
8.4 Electronic Design Automation (EDA) Tools	248
8.5 How to use the OrCAD EDA Tool?	249
8.6 Schematic Design using OrCAD Capture CIS	249
8.7 The PCB Layout Design	267
8.8 Printed Circuit Board (PCB) Fabrication	288
Summary	294
Keywords	294
Objective Questions	296
Review Questions	298
Lab Assignments	299
9. Embedded Firmware Design and Development	302
9.1 Embedded Firmware Design Approaches	303
9.2 Embedded Firmware Development Languages	306
9.3 Programming in Embedded C	318
Summary	371
Keywords	372
Objective Questions	373
Review Questions	378
Lab Assignments	380
10. Real-Time Operating System (RTOS) based Embedded System Design	381
10.1 Operating System Basics	382
10.2 Types of Operating Systems	386
10.3 Tasks, Process and Threads	390
10.4 Multiprocessing and Multitasking	402
10.5 Task Scheduling	404
10.6 Threads, Processes and Scheduling: Putting them Altogether	422
10.7 Task Communication	426
10.8 Task Synchronization	437
10.10 How to Choose an RTOS	470
Summary	480
Keywords	481
Objective Questions	483

	<u>Review Questions</u> 69	
	<u>Lab Assignments</u> 71	
3.	Characteristics and Quality Attributes of Embedded Systems	72
3.1	Characteristics of an Embedded System	72
3.2	Quality Attributes of Embedded Systems	74
4.	Embedded Systems—Application- and Domain-Specific	83
4.1	Washing Machine—Application-Specific Embedded System	83
4.2	Automotive—Domain-Specific Examples of Embedded System	85
	<u>Summary</u> 89	
	<u>Keywords</u> 90	
	<u>Objective Questions</u> 90	
	<u>Review Questions</u> 91	
5.	Designing Embedded Systems with 8bit Microcontrollers—8051	92
5.1	Factors to be Considered in Selecting a Controller	93
5.2	Why 8051 Microcontroller	94
5.3	Designing with 8051	94
5.5	8051/52 Variants	155
6.	Programming the 8051 Microcontroller	164
7.	Hardware Software Co-Design and Program Modelling	204
7.1	Fundamental Issues in Hardware Software Co-Design	205
7.2	Computational Models in Embedded Design	207
7.3	Introduction to Unified Modelling Language (UML)	214
7.4	Hardware Software Trade-offs	219
	<u>Summary</u> 220	



ft.	An Introduction to Embedded System Design with VxWorks and MicroC/OS-II RTOS	498
	11.1 VxWorks 499	
	11.2 MicroC/OS-II 514	
12.	lakBr4oo«ad1esUaeAT' Bed R«Wzr mdRmoarr	548
	IAI ffizqgznpjioj oFHazdwureundFirmrzgr< W	
	12.2 Board Power Up 553	
	Summary 554	
13.	The Embedded System Development Environment	556
	13.1 The Integrated Development Environment (IDE) 557	
	13.2 Types of Files Generated on Cross-compilation 588	
	13.3 Disassembler/Decompiler 597	
	13.5 Target Hardware Debugging 606	
	<i>Review Questions</i> 612	
14.	Product Enclosure Design and Development	Gl:g
	14.1 Product Enclosure Design Tools 616	
	14.2 Product Enclosure Development Techniques 616	
	<i>Review Questions</i> 620	
IS.	The Embedded Product Development Life Cycle (EDLC)	édl
	15.1 What is EDLC? 622	
	15.2 Why EDLC 622	
	15.3 Objectives of EDLC 623	
	15.4 Different Phases of EDLC 623	

15.5 EDLC Approaches (Modeling the EDLC) 636

16. Trends in the Embedded Industry 645

16.1 Processor Trends in Embedded System 645

16.5 Bottlenecks 652

TnrrOttariiOTiQJPQ Vemily OfMJUzgi;irKzjjElCrk NJ
tncrndzariinnlxjAS'RgFer«ilyn£kfcrg<\;<tlry1lcw rL57
fnrcmJzariizvil:+iABM@fiaioiLyeiiPrtxre seas dfid'

1. Digital Clock 669
2. Battery-Operated Smartcard Reader 699
3. Automated Meter Reading System (AMR) 701
4. Digital Camera 703



(R17A0464) EMBEDDED SYSTEMS

COURSE

OBJECTIVES:

For embedded systems, the course will enable the students to:

- 1) To understand the basics of microprocessors and microcontrollers architecture and its functionalities
- 2) Understand the core of an embedded system
- 3) To learn the design process of embedded system applications.
- 4) To understand the RTOS and inter-process communication.

UNIT-I: INTRODUCTION TO MICROPROCESSORS AND MICROCONTROLLERS: 8086

Microprocessor:

Architecture of 8086, Register Organization, Programming Model, Memory Segmentation, Signal descriptions of 8086, Addressing modes, Instruction Set.

8051 Microcontroller: 8051 Architecture, I/O Ports, Memory Organization, Instruction set of 8051.

UNIT-II: INTRODUCTION TO EMBEDDED SYSTEMS:

History of embedded systems, Classification of embedded systems based on generation and complexity, Purpose of embedded systems, Applications of embedded systems, and characteristics of embedded systems, Operational and Non-operational attributes of embedded systems.

UNIT-III: TYPICAL EMBEDDED SYSTEM

Core of the embedded system, Sensors and actuators, Onboard communication interfaces-I2C, SPI, parallel interface; External communication interfaces-RS232, USB, infrared, Bluetooth, Wi-Fi, ZigBee, GPRS.

UNIT-IV: EMBEDDED FIRMWARE DESIGN AND DEVELOPMENT: Embedded firmware design approaches-super loop based approach, operating system based approach; embedded firmware development languages-assembly language based development, high level language based development.

UNIT-V EMBEDDED PROGRAMMING CONCEPTS

Data types, Structures, Modifiers, Loops and Pointers, Macros and Functions, object oriented Programming, Embedded Programming in C++ & JAVA

TEXT BOOKS:

1. Embedded Systems, Raj Kamal, Second Edition TMH.
2. Kenneth. J. Ayala, The 8051 Microcontroller, 3rd Ed., Cengage Learning
3. Introduction to Embedded Systems - shibu k v, Mc Graw Hill Education.

REFERENCE BOOKS:

1. Advanced Microprocessors and Peripherals – A. K. Ray and K.M. Bhurchandi, TMH, 2nd Edition 2006
2. Embedded Systems- An integrated approach - Lyla B Das, Pearson education 2012.

COURSE OUTCOMES:

After going through this course the student will be able to

- 1) The student will learn the internal organization of popular 8086/8051 microprocessors/microcontrollers.
 - 2) Understand and design the Embedded systems
 - 3) Understand Embedded Firmware design approaches
 - 4) Learn the basics of RTOS
-

III B.Tech II Semester 2019-2020

SOFTWARE TESTING METHODOLOGIES(R17A0520)

TEXTBOOKS:

T1:Software Testing techniques – Baris Beizer, Dreamtech, second edition.

T2: Software Testing Tools – Dr.K.V.K.K.Prasad, Dreamtech

REFERENCES:

1. The craft of software testing – Brian Marick, Pearson Education.
2. Software Testing Techniques – SPD(Oreille)
3. Software Testing in the Real World – Edward Kit, Pearson.
4. Effective methods of Software Testing, Perry, John Wiley.
5. Art of Software Testing – Meyers, John Wiley.

SYLLABUS COVERAGE

UNITS	Topics	Course Learning Outcomes	Reference
UNIT-1	Introduction: Purpose of testing.	Explain the importance of testing and purpose of testing.	T1:1.1/covered
	Dichotomies, model for testing.	Illustrate different and compare dichotomies of testing.	T1:1.2/ covered
	Model for testing.	Demonstrate the model for testing and different testing levels and role of models.	T1:1.3/ covered
	Consequences of bugs, taxonomy of bugs.	Describe the consequences and taxonomy of bugs and different bugs in project environment.	T1:2.2/ covered
	Path testing and predicate, loops and path sensitization.	Illustrate the concepts of path testing and predicate loops and path sensitization.	T1:3.2/ covered
	Path instrumentation and their applications and link markers.	Explain Path instrumentation and their applications and link markers.	T1:3.5/ covered
Unit-2	Transaction flows techniques ,Transaction flows, transaction flow testing technique	List Transaction flows techniques and transaction flow structures and their test databases.	T1:4.3/ covered
	Basics of dataflow testing, strategies in dataflow testing, application of dataflow testing.	State Basics of data flow testing and Strategies in data flow testing, applications of dataflow testing.	T1:5.2/ covered
Unit-3	Domains and paths, Nice and ugly domains, domain testing.	Describe Domains and paths and explain about domains and bugs and their tools effectiveness.	T1:6.2/ covered
	Domains and interfaces testing, domain and interface testing, domains and testability.	Demonstrate Domains and Interfaces testing .explain linearising transformation and coordinate transformation.	T1:6.5/ covered
Unit-4	Path products and path expression.	State Path products and path expression, different laws used in path testing.	T1:8.3/ covered
	Reduction procedure, applications, regular expressions and flow anomaly detection.	Demonstrate Reduction procedure and applications, Regular expressions and Flow anomaly detection.	T1:8.4/ covered
	Logic based testing and decision tables.	Describe Logic based testing and Decision tables and compare hardware and software testing.	T1:10.2/ covered
	Path expressions, k v charts, specifications.	Illustrate Path expression and KV Charts and their specifications.	T1:11.3/ covered

Unit-5	State, State Graphs and Transition testing : State graphs, good & bad state graphs, state testing, Testability tips. .	Explain about the state testing and Difference between Good graph and bad graph	T1:12.2/ covered
	Graph Matrices and Applications: Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm	Demonstrate Reduction procedure, Power of matrix	T1:12.6/ covered
	Building tools		T1:12.7/NOT COVERED

(R17A0520) SOFTWARE TESTING METHODOLOGIES

Objectives:

- I. Be familiar with the concept of software testing objectives, process criteria, strategies and methods.
- II. Master various software testing issues and solutions in software unit test, integration, regression and system testing.
- III. Be familiar with the advanced software testing topics such as object oriented software testing methods and component based software testing issues, challenges and solutions.
- IV. Master the techniques and skills on how to use modern software testing tools to support Software testing projects.
- V. Be familiar with the important concepts of complexity metrics and object oriented metrics.
Be familiar with the knowledge of the foundations, techniques, and tools in area of software testing and its practice in the industry

UNIT-I:

Introduction:- Purpose of testing, Dichotomies, model for testing, consequences of bugs, taxonomy of bugs

Flow graphs and Path testing:- Basics concepts of path testing, predicates, path predicates and achievable paths, path sensitizing, path instrumentation, application of path testing.

UNIT-II:

Transaction Flow Testing:-transaction flows, transaction flow testing techniques. Dataflow testing:- Basics of dataflow testing, strategies in dataflow testing, application of dataflow testing.

UNIT-III:

Domain Testing:-domains and paths, Nice & ugly domains, domain testing, domains and interfaces testing, domain and interface testing, domains and testability.

UNIT-IV:

Paths, Path products and Regular expressions:- path products & path expression, reduction procedure, applications, regular expressions & flow anomaly detection.

Logic Based Testing:- overview, decision tables, path expressions, kv charts, specifications.

UNIT-V:

State, State Graphs and Transition testing:- state graphs, good & bad state graphs, state testing, Testability tips.

Graph Matrices and Application:-Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm, building tools. (Student should be given an exposure to a tool like JMeter or Win-runner).

TEXT BOOKS :

1. Software Testing techniques – Baris Beizer, Dreamtech, second edition.
2. Software Testing Tools – Dr.K.V.K.K.Prasad, Dreamtech.

III-Year II-Sem B.Tech CSE 2019-20

PYTHON PROGRAMMING(R17A0554)

TEXT BOOKS

- T1. Allen B. Downey, ``Think Python: How to Think Like a Computer Scientist,,,,, 2nd edition,
Updated for Python 3, Shroff/O,,Reilly Publishers, 2016.
- T2. R. Nageswara Rao, “Core Python Programming”, dreamtech
- T3. Python Programming: A Modern Approach, Vamsi Kurama, Pearson

SYLLABUS COVERAGE

Unit	Topics	Course Learning Outcomes	Page Reference
I	Introduction to Python and installation	Learn the installation of Python	T2:2-3 T2:19-21
	data types: Int, float, Boolean, string, and list	Understand data types in Python	T2:51-53 T2:55,56-60
	variables, expressions, statements , precedence of operators, comments	Explain about the Operators	T2:46,49 T2:71-87
	functions--- function and its use, flow of execution, parameters and arguments.	Define a function	T2:237-238 T2:253-256
II	Conditionals: Boolean values and operators, conditional (if), alternative (if-else), chained conditional (if-elif-else); Iteration: while, for, break, continue.	Explain about types of conditional statements	T2:118-139
III	Fruitful functions: return values, parameters, local & global scope , function composition, recursion;	Learn the concept of functions	T2:238-262
	Strings: string slices, immutability, string functions and methods, string module;	Define a string. Learn the string methods	T2:207-225
	Python arrays, Access the Elements of an Array, array methods.	Defining an array and its methods	T2:151-160
IV	Lists: list operations, list slices, list methods, mutability, aliasing , cloning lists, list parameters, list comprehension;	Define a List. Learn the List methods	T2:283-293 T2:305
	Tuples: tuple assignment, tuple as return value, tuple comprehension;	Define a tuple. Learn the tuple methods	T2:307-311
	Dictionaries: operations and methods	Define a dictionary. Learn the dictionary methods	T2:322-324
	Files and exception: text files, reading and writing files,	Understand working with files in python	T2:441-446
	command line arguments	Learn to use command line arguments	T2:108-110
	errors and exceptions, handling exceptions	Learn to handle exceptions in python	T2:421-429

V	modules (datetime, time, OS , calendar, math module), Explore packages.	Understand working with different modules in python	T2:515-519 T2:532,588
---	---	---	--------------------------

(R17A0554) PYTHON PROGRAMMING

COURSE OBJECTIVES:

- To read and write simple Python programs.
- To develop Python programs with conditionals and loops.
- To define Python functions and call them.
- To use Python data structures — lists, tuples, dictionaries.
- To do input/output with files in Python.

UNIT I

INTRODUCTION DATA, EXPRESSIONS, STATEMENTS Introduction to Python and installation, data types: Int, float, Boolean, string, and list; variables, expressions, statements, precedence of operators, comments; modules, functions--- function and its use, flow of execution, parameters and arguments.

UNIT II

CONTROL FLOW, LOOPS Conditionals: Boolean values and operators, conditional (if), alternative (if- else), chained conditional (if-elif-else); Iteration: while, for, break, continue.

UNIT III

FUNCTIONS, ARRAYS Fruitful functions: return values, parameters, local and global scope, function composition, recursion; Strings: string slices, immutability, string functions and methods, string module; Python arrays, Access the Elements of an Array, array methods.

UNIT IV

LISTS, TUPLES, DICTIONARIES Lists: list operations, list slices, list methods, list loop, mutability, aliasing, cloning lists, list parameters, list comprehension; Tuples: tuple assignment, tuple as return value, tuple comprehension; Dictionaries: operations and methods, comprehension;

UNIT V

FILES, EXCEPTIONS, MODULES, PACKAGES Files and exception: text files, reading and writing files, command line arguments, errors and exceptions, handling exceptions, modules (datetime, time, OS , calendar, math module), Explore packages.

COURSE OUTCOMES:

Upon completion of the course, students will be able to Read, write, execute by hand simple Python programs. • Structure simple Python programs for solving problems. • Decompose a Python program into functions. • Represent compound data using Python lists, tuples, dictionaries. • Read and write data from/to files in Python Programs •

TEXT BOOKS

- 1.Allen B. Downey, ``Think Python: How to Think Like a Computer Scientist,,,,, 2nd edition, Updated for Python 3, Shroff/O,,Reilly Publishers, 2016.
- 2.R. Nageswara Rao, “Core Python Programming”, dreamtech
3. Python Programming: A Modern Approach, Vamsi Kurama, Pearson

REFERENCE BOOKS: 1. Core Python Programming, W.Chun, Pearson. 2. Introduction to Python, Kenneth A. Lambert, Cengage 3. Learning Python, Mark Lutz, Orielly

III B.Tech II Semester

Subject: WEB TECHNOLOGIES

Branch: CSE

SubjectCode: R17A0519

AcademicYear:2019-2020 TEXT BOOKS:

1. Web Technologies, Uttam K Roy, Oxford University Press
2. The Complete Reference PHP – Steven Holzner, Tata McGraw-Hill

References:

1. Programming world wide web-Sebesta, Pearson Education ,2007.
2. Core SERVLETS ANDJAVASERVER PAGES VOLUME 1: CORE TECHNOLOGIES By Marty Hall and Larry Brown Pearson
3. Internet and World Wide Web – How to program by Dietel and Nieto PHI/Pearson Education Asia.
4. Jakarta Struts Cookbook, Bill Siggelkow, S P D O'Reilly for chap 8.
5. March's beginning JAVA JDK 5, Murach, SPD
6. An Introduction to Web Design and Programming –Wang-Thomson

SYLLABUS COVERAGE

UNIT	Topic	Course Learning Outcomes	Referen ces
UNIT-1	Web Basics and Overview	Analazing the WWW and internet	T1
	HTML Common tags	Creating the static webpages	T1
	CSS	Illustrate the web page with CSS	T1
	Client side Scripting: Introduction to Java	Creating a Javascript program	T1
	Java script language- declaring variables		T1
	Scope of variables, Functions		T1
	Event Handlers (onclick, onsubmit etc.)	Demonstrating event handlers	T1
	Form Validation	Illustrate the client side validation	T1
UNIT-2	Introduction to PHP, declaring variables, data types	Building a web application using PHP on a web server	T2
	Operators, expressions, control structures		T2

	functions		T2
	Reading data from controls like text boxes, check boxes, radio buttons, lists etc.	Demonstrating reading data from form fields and files	T2
	Handling file uploads		T2
	Handling results, Handling Sessions and Cookies	Explain creation of cookies and sessions	T2
	XML: Introduction to XML, defining XML	Defining XML and differentiate with HTML	T1
	Document type definition	Explain the importance of DTD	T1
	XML Schemas, Document Object Model	Defining XML schema	T1
	XHTML	Differentiate XHTML and html	T1
	Parsing XML Data: DOM and SAX Parsers	Describing parsing of data using parsers	T1
UNIT-3	Introduction to Servlets:	Demonstrating servlet	T1
	Life cycle of Servlet, deploying a Servlet	Illustrating a servlet program deployment on a webserver	T1
	The Servlet API	Demonstrate different classes and interfaces of servlet	T1
	Reading Servlet parameters	Create a servlet app that reads the parameters	T1
	Handling Http Request & Responses	Build web application for request and response objects	T1
	Using Cookies and Sessions		T1
	Example Programs		T1
UNIT-4	Introduction to JSP: The Anatomy of a JSP	Differentiate Servlet and JSP	T1
	JSP Processing	Demonstrate the JSP processing	T1
	Declarations, Directives, Expressions	Classifying the JSP elements	T1
	Code Snippets, input Objects	Creating applications using impl. Objects	T1
	Using Cookies and Sessions for Session handling	Demonstrate cookies and sessions using JSP	T1
UNIT-5	Fundamentals of JDBC and its importance, uses, strengths and weaknesses.	Demonstrate Database access	T1
	JDBC Drivers	Classifying the JDBC drivers	T1
	Javax.sql.* package	Explain the classes and interfaces of JDBC	T1
	Connecting to a servlets using JDBC	Creating a JDBC app	T1
	Connecting to database in JSP	Build a JSP with JDBC	T1
	Connecting to PHP, Executing simple queries	Creating PHP application for storing and retrieving data	T1

SYLLABUS

(R17A0519) WEB TECHNOLOGIES

Objectives:

- Giving the students the insights of the Internet programming and how to design and implement complete applications over the web.
- It covers the notions of Web servers and Web Application Servers, Design Methodologies with concentration on Object-Oriented concepts
- Programming, Server-Side Programming, Active Server Pages, Database Connectivity to web applications, Adding Dynamic content to web applications,
- Programming Common Gateway Interfaces, Programming the User Interface for the web applications.

UNIT I:

Web Basics and Overview: Introduction to Internet, World Wide Web, Web Browsers, URL, MIME, HTTP, Web Programmers Tool box.

HTML Common tags: List, Tables, images, forms, frames, Basics of CSS and types of CSS.

Client-Side Programming (Java Script): Introduction to Java Script, declaring variables, functions, Event handlers (onclick, onsubmit, etc.,) and Form Validation.

UNIT II:

Server-Side Programming (PHP): Declaring Variables, Data types, Operators, Control structures, Functions, Reading data from web form controls like text buttons, radio buttons, list, etc., Handling File Uploads, Handling Sessions & Cookies.

Introduction to XML: Document type definition, XML Schemas, Document Object model, Presenting XML , Introduction to XHTML, Using XML Processors: DOM and SAX.

UNIT III:

Web Servers and Servlets: Tomcat web server, Installing the Java Software Development Kit, Tomcat Server & Testing Tomcat.

Introduction to Servlets: Lifecycle of a Servlet, JSDK, Deploying Servlet, The Servlet API, The javax. Servlet Package, Reading Servlet parameters, Reading Initialization parameters. The javax.servlet HTTP package, Handling Http Request & Responses, Using Cookies-Session Tracking.

UNIT IV:

Introduction to JSP: The Problem with Servlet. The Anatomy of a JSP Page, JSP Processing. JSP Application Design with MVC Setting Up and JSP Environment, JSP Declarations, Directives, Expressions, Code Snippets, implement objects, Requests, Using Cookies and Session for Session Tracking.

UNIT V:

Database Access: Database Programming using JDBC, JDBC drivers, Studying javax.sql.* package, Connecting to database in PHP, Execute Simple Queries, Accessing a Database from a Servlet and JSP page.

TEXT BOOKS:

3. Web Technologies, Uttam K Roy, Oxford University Press
4. The Complete Reference PHP – Steven Holzner, Tata McGraw-Hill

REFERENCE BOOKS:

7. Programming world wide web-Sebesta, Pearson Education ,2007.
8. Core SERVLETS AND JAVASERVER PAGES VOLUME 1: CORE TECHNOLOGIES By Marty Hall and Larry Brown Pearson
9. Internet and World Wide Web – How to program by Dietel and Nieto PHI/Pearson Education Asia.
10. Jakarta Struts Cookbook, Bill Siggelkow, S P D O'Reilly for chap 8.
11. March's beginning JAVA JDK 5, Murach, SPD
12. An Introduction to Web Design and Programming –Wang-Thomson

OUTCOMES:

- Analyze a web page and identify its elements and attributes.
- Create web pages using XHTML and Cascading Styles sheets.
- Demonstrate knowledge on Installation and usage of Server software's.
- Understand Database Connectivity to web applications
- Build web applications using Servlet and JSP