MIC-3	One from the pool								4					100
Multidisciplinary Course(s)														
MDC-3	One from the pool								3					75
Ability Enhancement Course(s)														
AEC-3	One from the pool								2					50
Total Credits									**20**					**500**

Semester 4

Course Code	**Course Title**	**CourseID**	**L**	**T**	**P**	**L**	**T**	**P**	**Credits**	**MARKS**				
			(Hrs)			**Credits**				**TI**	**TE**	**PI**	**PE**	**Total**
Core Course(s)														
CC-ID10	Computer Networks	240/BCA/CC401	3	-	2	3	-	1	4	25	50	05	20	100
CC-ID11	Software Engineering	240/BCA/CC402	3	1	-	3	1	-	4	30	70	--	--	100
CC-ID12	Java Programming	240/BCA/CC403	3	-	2	3	-	1	4	25	50	05	20	100
Minor/ Vocational Course(s)														
MIC-4	One from the pool								4					100
Ability Enhancement Course(s)														
AEC-4	One from the pool								2					50
Value-added Course(s)														
VAC-3	One from the pool								2					50
Total Credits									**20**					**500**

Semester 5

Course Code	**Course Title**	**Course ID**	**L**	**T**	**P**	**L**	**T**	**P**	**Credits**	**MARKS**				
			(Hrs)			**Credits**				**TI**	**TE**	**PI**	**PE**	**Total**
Core Course(s)														
CC-ID13	Computer Graphics	240/BCA/CC501	3	-	2	3	-	1	4	25	50	05	20	100
CC-ID14	Python Programming	240/BCA/CC502	3	-	2	3	-	1	4	25	50	05	20	100

SEMESTER 4

240/BCA/CC401

Course code	**CC-ID10**			
Category	Core Course			
Course title	**Computer Networks**			
Scheme and Credits	L	T	P	Credits
	3	0	2	4
Theory Internal	25			
Theory External	50			
Practical Internal	05			
Practical External	20			
Total	100			
Duration of Exam	3 Hrs			

Note: The examiner will set nine questions in total. Question one will have seven parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each attempt FIVE questions in all, selecting one question from every unit apart from the Question Number 1.

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

CO1: Explain the functions of the different layers of the OSI Protocol.

CO2: Draw the functional block diagram of wide-area networks (WANs), local area networks (LANs), and Wireless LANs (WLANs) and describe the function of each.

CO3: Identify and connect various connecting components of a computer network.

CO4: Configure DNS DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, and Firewalls using open-source available software and tools.

CO5: outline various models, topologies and devices of Computer Networks.

UNIT - I

Introduction: Data communication, Components, Data Representation, Simplex, Half Duplex, and Full Duplex Transmission, Modulation, Multiplexing, Computer networks, distributed processing, Internet, Topologies, Packet and circuit switching, connectionless and connection- oriented services. Network Models: OSI model and TCP/IP Model Physical Layer – LAN: Ethernet.

UNIT – II

Data Link Layer and Medium Access Sub Layer: MAC Addressing, Framing, Stop and Wait, Go back – N ARQ, Selective Repeat ARQ, Sliding Window Protocol. Medium Access Control: Random access, Controlled Access, and channelization protocols. Network Layer: Logical

addressing, classful and classless addressing, subnetting, Ipv4, ICMPv4, ARP, RARP and BOOTP, Ipv6, Ipv6 addressing.

UNIT - III

Network Devices: Repeater, hub, switch, router, and gateway. Routing Algorithms: introduction to routing, Shortest Path Algorithm, Flooding, Hierarchical Routing, Link State, and Distance Vector Routing Transport Layer: Process to Process Communication, User Datagram Protocol (UDP), Transmission Control Protocol (TCP), TCP connection management.

UNIT - IV

Congestion Control, Quality of Service, QoS Improving techniques. Application Layer: Domain Name Space (DNS), EMAIL, File Transfer Protocol (FTP), HTTP, SNMP Network Security: Firewalls, security goals, types of attack, symmetric and asymmetric key ciphers.

TEXT AND REFERENCE BOOKS:

1. Essentials of Computer And Network Technology, Nasib Singh Gill, Khanna Book Publishing Co.(P) Ltd, New Delhi.
2. Data and Computer Communication, 8th Edition, William Stallings, Pearson Prentice Hall India.
3. Computer Networks, latest Edition, Andrew S. Tanenbaum, Pearson New International Edition.
4. Data Communication and Networking, 4th Edition, Behrouz A. Forouzan, McGraw-Hill.
5. Internetworking with TCP/IP, Volume 1, latest Edition Douglas Comer, Prentice Hall of India.
6. TCP/IP Illustrated, Volume 1, W. Richard Stevens, Addison-Wesley, United States of America.

Computer Networks lab

List of Experiments

1. To construct a simple network topology on Packet Tracer.

2. To verify and configure VLAN and VLAN trunk in packet tracer.

3. To construct RJ45 cable.

4. a) To configure simple static routing.

b) To implement Security on interconnecting devices.

5. To configure a Network Topology constitutes Routers and Switches using Packet Tracer.

6 Working with complex network topologies.

7. Mid Term Evaluation 8. To monitor network traffic using Wire Shark

9. To get the MAC or Physical Address of the system Using Address Resolution Protocol.

10. To Configure network using Routing Information Protocol (RIP)

11. To configure network state routing protocol (OSPF).

12. To configure Border Gateway Protocol.

13. To configure Application Layer protocols: DHCP and DNS.

Course code	CC-ID11			
Category	Core Course			
Course title	**Software Engineering**			
Scheme and Credits	L	T	P	Credits
	3	1	0	4
Theory Internal	30			
Theory External	70			
Total	100			
Duration of Exam	3 Hrs			

Note: The examiner will set nine questions in total. Question one will have seven parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each attempt FIVE questions in all, selecting one question from every unit apart from the Question Number 1.

COURSE OUTCOMES:

At the end of this unit, students will demonstrate the ability to:

CO1: Understand the fundamental principles and characteristics of software, including its evolution and the evolving role of software in modern systems.

CO2: Apply basic concepts of Software Design, including Architectural Design, Modularization, Design Structure Charts, Flow Charts, and measures of Coupling and Cohesion.

CO3: Define Software Quality attributes and understand Software Quality Assurance plans and activities, including Software Documentation

CO4: Understand the need for Software Maintenance and differentiate between categories such as Preventive, Corrective, and Perfective Maintenance.

UNIT-I

Introduction: Software and its Characteristics, Evolving Role of Software, Software Product. Software Processes. Software Crisis. Software Engineering Evolution. Principles of Software Engineering. Programming-in-the-small vs. Programming-in-the-large. Software Components. Software Engineering Processes.

Software Life Cycle (SLC) Models: Water-Fall Model. Prototype Model. Spiral Model. Evolutionary Development Models. Iterative Enhancement Models. Object Oriented Models and other latest Models.

UNIT-II

Software Requirements: Functional and Non-Functional. User requirements. System requirements. Software Requirements Document - Requirement Engineering Process: Feasibility Studies, requirements validation, requirements management.

Software Design: Basic Concept of Software Design. Architectural Design. Low Level Design: Modularization. Coupling and Cohesion Measures; Design Strategies: Function Oriented Design. Object Oriented Design. Top• Down and Bottom-Up Design.

UNIT-III

Software Measurement and Metrics: Process Metrics. Project metrics. Estimation-LOC, Halstead's Software Science. Function Point (FP). Cyclomatic Complexity Measures: Software Project Estimation Models- Empirical. Putnam. COCOMO I & II.

Software Quality: Attributes, Software Quality Assurance - plans & activities: Software Documentation.

Software Testing: Introduction. Verification vs. Validation. Software Reliability. Levels of Testing.Structural Testing, Functional Testing.

UNIT-IV

Software Maintenance: Need for Maintenance. Categories of Maintenance: Preventive. Corrective and Perfective Maintenance. Cost of Maintenance: Software Re-Engineering. Reverse Engineering,Software Documentation.

Software Configuration Management: SCM Activities. Change Control Process. Software VersionControl: Software Reuse.

Textbooks & References:

1. Software Engineering, Nasib Singh Gill, Khanna Book Publishing Co.(P) Ltd, New Delhi.
2. Software Engineering – A Practitioner's Approach, Roger S. Pressman, 1996, MGH.
3. Fundamentals of software Engineering, Rajib Mall, PHI
4. Software Engineering by Ian Summerville, Pearson Edu, 5th edition, 1999, AW,
5. Software Engineering – David Gustafson, 2002, T.M.H
6. Software Engineering Fundamentals Oxford University, Ali Behforooz and Frederick J. Hudson 1995 JW&S,
7. An Integrated Approach to software engineering by Pankaj jalote , 1991 Narosa

240/BCA/CC403

Course code	**CC-ID12**			
Category	Core Course			
Course title	**Java Programming**			
Scheme and Credits	L	T	P	Credits
	3	0	2	4
Theory Internal	25			
Theory External	50			
Practical Internal	05			
Practical External	20			
Total	100			
Duration of Exam	3Hrs			

Note: The examiner will set nine questions in total. Question one will have seven parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each attempt FIVE questions in all, selecting one question from every unit apart from the Question Number 1.

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

CO1: Identify classes, objects, members of a class and relationships among them for a specific problem.

CO2: Understand and demonstrate the concepts of garbage collection, polymorphism, inheritance etc.

CO3: Do numeric (algebraic) and string-based computation.

CO4: Understand and implement modularity as well as basic error-handling techniques.

CO5: Develop, design and implement small multithreaded programs using Java language.

UNIT - I

Introduction to Java: Evolution of Java, Object Oriented Programming Structure, Overview and characteristics of Java, Java program Compilation and Execution Process, Organization of the Java Virtual Machine, Client side Programming, Platform Independency & Portability, Security, Relation b/w JVM, JRE and JDK, Introduction to JAR format, Naming Conventions, Data types & Type casting, operators.

UNIT - II

OOPS Implementation: Classes, Objects, attributes, methods, data encapsulation, reference variables, Constructors, Anonymous block, Method Overloading, Static Data members.

Argument Passing Mechanism: Passing primitive arguments, passing objects, Wrapper Classes; This keyword: Referencing instance members, Intra class constructor chaining, Method chaining; Inheritance & code reusability: Extending classes for code reusability, Usage of super keyword, Method Overriding.

UNIT - III

Packages: Defining Package, CLASSPATH, Package naming, Accessibility of Packages, using Package Members.

Interfaces: Implementing Interfaces, Interface and Abstract Classes, Extends and Implements together.

Exceptions Handling: Exception, Handling of Exception, Using try-catch, Catching Multiple Exceptions, Using finally clause, Types of Exceptions, Throwing Exceptions, Writing Exception Subclasses.

UNIT - IV

Multithreading: Introduction, The Main Thread, Java Thread Model, Thread Priorities, Synchronization in Java, Inter thread Communication.

I/O in Java: I/O Basics, Streams and Stream Classes, The Predefined Streams, Reading from, and Writing to, Console, Reading and Writing File.

Strings and Characters: Fundamentals of Characters and Strings, The String Class, String Operations, String Buffer Class and Methods.

TEXT AND REFERENCE BOOKS:

1. E. Balaguruswamy, "Programming with Java", TMH
2. Horstmann, "Computing Concepts with Java 2 Essentials", John Wiley.
3. Decker &Hirshfield, "Programming.Java", Vikas Publication.
4. Patrick Naughton and HerbertzSchidt, "Java-2 the complete Reference", TMH
5. Sierra & bates, "Head First Java", O'Reilly.

Java Programming lab

List of Experiments

1. Write a java program to find the Fibonacci series using recursive and non recursive functions
2. Write a java program to multiply two given matrices.
3. Write a java program for Method overloading and Constructor overloading .
4. Write a program to demonstrate execution of static blocks, static variables & static methods.
5. Write a program to display the employee details using Scanner class
6. Write a program for sorting a given list of names in ascending order
7. Write a program to implement single and Multi level inheritance
8. Write a program to implement Hierarchical Inheritance.
9. Write a program to implement method overriding.
10. Write a program to create an abstract class named Shape that contains two integers and an empty method named printArea (). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method printArea () that prints the area of the given shape.
11. Write a program to implement Interface.
12. Write a program to implement multiple and Hybrid Inheritance
13. Write a program to create inner classes
14. Write a program to create user defined package and demonstrate various access modifiers.
15. Write a program to demonstrate the use of super and final keywords.
16. Write a program if number is less than 10 and greater than 50 it generate the exception out of range. Else it displays the square of number.
17. Write a program with multiple catch Statements.
18. Write a Program to implement simple Thread by extending Thread class and implementing runnable interface.
19. Write a program that implements a multi-thread application that has three threads