BMS COLLEGE OF ENGINEERING BENGALURU-19
(Autonomous College under VTU)
Department of Computer Applications
SEMESTER –III

COURSE TITLE	Computer Networks	**Credits**	5
COURSE CODE	16MCA3DCN1	**L-T-P-S**	4-0-1-0
CIE	50	**SEE**	50

Prerequisites: None

UNIT 1:

Foundation: Applications, Requirements, Network Architecture, Implementing Network Software, Performance, Network Hardware, Network Software. **(09 Hrs.)**

UNIT 2:

Reference Model: OSI and TCP/IP and their Comparison, Guided Transmission Media, Digital Modulation and Multiplexing. **(09 Hrs.)**

UNIT 3:

Data link layer design issues, Error Detection and Correction, Elementary Data Link Protocols, Sliding Window Protocols, The Channel: Allocation Problem, Multiple Access Protocols, and Wireless LANs. **(10 Hrs.)**

UNIT 4:

Routing Algorithms: The Shortest Path Algorithm, Flooding, Distance Vector Routing, Link State Routing and Hierarchical Routing, Congestion Control Algorithms, Quality of Service: Application Requirements, Traffic Shaping, Packet Scheduling, The Network Layer in the Internet: The IP Version 4 Protocol, IP Addresses, IP Version 6, Internet Control Protocols. **(10 Hrs.)**

UNIT 5:

Elements of Transport Protocols, The Internet Transport Protocols: UDP, TCP: Introduction to TCP, The TCP Service Model, The TCP Protocol, The TCP Segment Header, TCP Connection Establishment, TCP Connection Release. **(10 Hrs.)**

Text Books:

Sl. No.	**Content**
1.	Andrew S. Tanenbaum and David J. Wetherall, “Computer Networks”, 5th edition, Prentice Hall, 2014.
2.	Larry L Peterson and Bruce S. Davie, "Computer Networks”: A Systems Approach 6th Edition, Morgan Kaufmann. 2016.

Reference Books:

Sl. No.	Content
1.	Kurose and Ross “Computer Networking”: A Top-Down Approach (6th or 7th Editions), Pearson Publication 2016.
2.	Cisco Networking Academy Program, CCNA 1 and 2 Companion Guide 2016.

List of Lab Programs – Integrated with Computer Network Theory:

1. Setting up networks using a simulator [Wired and Wireless].
2. Implementation of Channel Allocation Problem.
3. Implementation of Routing Algorithm.
4. Implementing of TCP and UDP Protocols.

Course Outcomes:

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

CO1	Describe the concepts of Computer Networks.
CO2	Identify the effect of network devices on health and environment.
CO3	Apply the concepts of Computer network for a given problem.
CO4	Demonstrate the concepts of Computer Network.

BMS COLLEGE OF ENGINEERING BENGALURU-19

(Autonomous College under VTU)

Department of Computer Applications

SEMESTER – III

COURSE TITLE	Java Programming - I	**Credits**	5
COURSE CODE	16MCA3DCJ1	**L-T-P-S**	4-0-1-0
CIE	50	**SEE**	50

Prerequisites: 16MCA2DCOP - Object Oriented Programming with C++

UNIT 1:

The History and Evolution of Java, Java's Lineage, The Birth of Modern Programming: C, C++: The Next Step The Stage is Set for Java, The Creation of Java, How Java Changed the Internet, Security, Portability, Java's Magic: The Bytecode, The Java Buzzwords: Simple, Object-Oriented, Robust, Multithreaded, Architecture-Neutral, Interpreted and High Performance, Distributed, Dynamic, The Evolution of Java SE 8, A Culture of Innovation.

An Overview of Java, Object-Oriented Programming, Two Paradigms, Abstraction, The Three OOP Principles, A First Simple Program, Entering the Program, Compiling the Program, A Closer Look at the First Sample Program, A Second Short Program.

Introducing Classes, Class Fundamentals, The General Form of a Class, A Simple Class, Declaring Objects, A Closer Look at new, Assigning Object Reference Variables, Introducing Methods: Adding a Method to the Box Class, Returning a Value, Adding a Method That Takes Parameters, Constructors, Parameterized Constructors, this Keyword, Instance Variable Hiding, Garbage Collection, The finalize( )Method. **(09 Hrs.)**

UNIT 2:

A Closer Look at Methods and Classes, Overloading Methods, Overloading Constructors

Using Objects as Parameters, A Closer Look at Argument Passing, Returning Objects, Recursion, Introducing Access Control, Understanding static, Introducing final, Arrays Revisited, Introducing Nested and Inner Classes, Exploring the String Class, Using Command-Line Arguments, Varargs: Variable-Length Arguments, Overloading Vararg Methods, Varargs and Ambiguity.

Inheritance, Inheritance Basics, Member Access and Inheritance, A More Practical Example, A Superclass Variable Can Reference a Subclass Object, Using super, Using super to Call Superclass Constructors, A Second Use for super, Creating a Multilevel Hierarchy, When Constructors Are Executed, Method Overriding, Dynamic Method Dispatch, Why Overridden Methods, Applying Method Overriding, Using Abstract Classes, Using final with Inheritance, Using final to Prevent Overriding, Using final to Prevent Inheritance, The Object Class, String Handling. **(09 Hrs.)**

UNIT 3:

Packages and Interfaces: Packages, Defining a Package, Finding Packages and CLASSPATH, A Short Package Example, Access Protection, An Access Example, Importing Packages, Interfaces, Defining an Interface, Implementing Interfaces, Nested Interfaces, Applying Interfaces, Variables in Interfaces, Interfaces Can Be Extended, Default Interface Methods, Default Method Fundamentals, A More Practical Example,

Multiple Inheritance Issues, Use static Methods in an Interface, Final Thoughts on Packages and Interfaces.
Exception Handling, Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, Displaying a Description of an Exception, Multiple catch Clauses, Nested try Statements, throw, throws, finally, Java's Built-in Exceptions, Creating Your Own Exception Subclasses, Chained Exceptions, Three Recently Added Exception Features, Using Exceptions.

Multithreaded Programming: The Java Thread Model, Thread Priorities, Synchronization Messaging, The Thread Class and the Runnable, The Main Thread, Creating a Thread, Implementing Runnable, Extending Thread, Choosing an Approach, Creating Multiple Threads, Using is Alive( ) and join( ), Thread Priorities, Synchronization, Using Synchronized Methods, The synchronized Statement, Interthread Communication, Deadlock , Suspending, Resuming and Stopping, Obtaining A Thread's, Using Multithreading. **(10 Hrs.)**

UNIT 4:

Enumerations, Autoboxing, Enumeration Fundamentals, The values ( ) and value Of( ) Methods, Java Enumerations Are Class Types, Enumerations Inherit Enum, Another Enumeration Example, Type Wrappers, Character, Boolean, The Numeric Type Wrappers, Autoboxing, Autoboxing and Methods, Autoboxing / Unboxing Occurs in Expressions ,Autoboxing / Unboxing Boolean and Character Values, Autoboxing / Unboxing Helps Prevent Errors, A Word of Warning.
I/O: I/O Basics Streams, Byte Streams and Character Streams, The Predefined Streams, Reading Console Input, Reading Characters, Reading Strings, Writing Console Output, The PrintWriter Class, Reading and Writing Files.
Generics: What Are Generics, A Simple Generics Example, Generics Work Only with Reference Types, Generic Types Differ Based on Their Type Arguments, How Generics Improve Type Safety, A Generic Class with Two Type Parameters, The General Form of a Generic Class, Bounded Types, Using Wildcard Arguments, Bounded Wildcards, Creating a Generic Method, Generic Constructors, Generic Interfaces, Raw Types and Legacy Code.
Lambda Expressions: Introducing Lambda Expressions, Lambda Expression Fundamentals, Functional Interfaces, Some Lambda Expression Examples, Block Lambda Expressions. **(10 Hrs.)**

UNIT 5:

The Collections Framework, Collections Overview, JDK 5 Changed the Collections Framework, Generics Fundamentally Changed the Collections Framework, Autoboxing Facilitates the Use of Primitive Types, the For-Each Style for Loop, the Collection Interfaces: The Collection Interface, The List Interface, The Set Interface , The Sorted Set Interface, The Navigable Set Interface, The Queue Interface, The Deque Interface, The Collection Classes, The Array List Class, The Linked List Class, The Hash Set Class, The Linked Hash Set Class, The Tree Set Class, The Priority Queue Class, The Array Deque Class, The Enum Set Class, Accessing a Collection via an Iterator, Using an Iterator, Vector, Stack.
Input/Output: Exploring java.io, Exploring NIO, Networking, The Applet Class, Event Handling, Introducing the AWT: Working with Windows, Graphics, and Text, Using AWT Controls, Layout Managers, and Menus, Introducing Swing, Exploring Swing, Introducing Swing Menus. **(10 Hrs.)**

Text Books:

Sl. No.	Content
1.	Herbert Schildt, "Java The Complete Reference", 9th Edition, Comprehensive Coverage of Java Language, Oracle Press, McGraw Hill Education (India Edition) 2014.

Reference Books:

Sl. No.	Content
1.	C Xavier, "Java Programming – A Practical Approach", Tata McGraw Hill, 2011
2.	Hari Mohan Pandey, "Java Programming", 1st Impression, Pearson, 2012
3.	Ralph Bravaco, Shai Simonson, "Java Programming – From the Ground Up", Tata McGraw Hill 2012.
4.	T V Suresh Kumar, B Eshwara Reddy, P Raghavan, "Programming with Java", Sanguine Technical Publishers, 2011.

List of Lab Programs – Integrated with Programming with Java Programming Theory:

1. Write a program that simulates a simple ATM machine based on transaction code entered by customer(USING else –if)
   1 – Withdrawal
   2 – Deposit
   3 – Check Balance
   4 – Exit.

2. Write a program that reads a list of numbers representing deposits to and withdrawal from a savings account. Positive entries represent deposits and the negative entries withdrawals. The program should calculate the sum of all deposits and the sum of all withdrawals. Use the entry "0" as signal to end of data.

3. Create a class TV with the attribute channel, volume, and switch indicating whether TV is On / Off. The methods needs to define:
   a. Turn the TV ON or Off
   b. Set the channel number 0 to 99
   c. Raise the volume by one unit, Range from 0 to 20 d) View TV status (ON /OFF), channel and volume.

Newly TV object is set to off with the channel set to 2 and volume initially 10.\

4. Create an abstract class Accounts with members balance, account number, account holder Name, address and methods withdrawal (), deposit(), display(). Create a subclass Savings Account with rate of Interest, calculate amount (), display(). Create another subclass of Account class as Current Account with overdraft limit, display (). Create these objects call their methods using appropriate constructors.

5. Define Circle, Square, and Triangle classes each of which implements the Geometry interface. Interface contains two methods area() and perimeter() methods.

6. Develop any three utility methods and keep corresponding classes in a package named myutil.

7. Write a program for user defined exception that checks the internal and external marks; if the internal marks greater than 40 it raises the exception “Internal Marks is Exceed”; if the external marks is greater than 60 it raises the exception and displays the message “The External Marks is Exceed”. Create the above exception and implement in program.

8. Write a program using Synchronized Threads, which demonstrates Producer Consumer concept.

9. (a) Create an enumeration Day Of Week with seven values SUNDAY to STURDAY. Add a method is Work Day() to the Day Of Week class that returns true if the value on which it is called is MONDAY through FRIDAY. For example, the call DayOfWeek SUNDAY is Work Day () returns false.
   (b) Write a Java program to demonstrate Boxing and Unboxing concept.

10. (a)Write a program to display the names of companies whoever quoted the lower price for specific product. Input data consists of names of companies and their quoted price.
    (b) Develop a Java program to demonstrate the features of Generics.

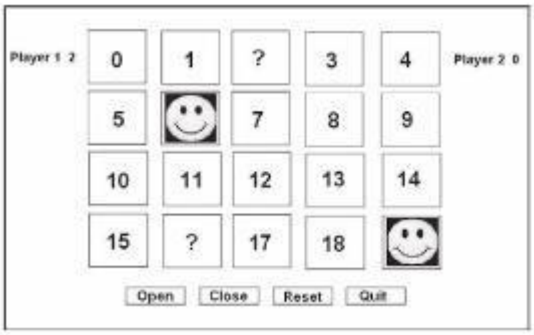
11. Write Java program for:
    a. Declare a string object named S1 containing a string “MCA BMSCE”
    b. Print the entire string
    c. Find length of the string
    d. Find the character position
    e. Print the words one after other in the given string.

(For example as per example String is “MCA BMSCE”, it should print MCA BMSCE)

12. Write a java program which demonstrates the utilities of Linked List class.

13. Write a program that displays two buttons at the bottom of a frame: one reads STOP and the other GO. When STOP is clicked, the application should display a red circle above the buttons, and when GO is clicked, a green circle.

14. Create a frame shown below.



Course Outcomes:

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

CO1	Identify classes, objects, members of a class and the relationships among them for a specific problem
CO2	Illustrate the concepts of String Handling and Exception handling.
CO3	Develop programs using the Java Collection API as well as the Java standard class library
CO4	Make use of the Swings in application development.

BMS COLLEGE OF ENGINEERING BENGALURU-19
(Autonomous College under VTU)
Department of Computer Applications
SEMESTER –III

COURSE TITLE	Object Oriented Modelling and Design	**Credits**	5
COURSE CODE	16MCA3DCOM	**L-T-P-S**	4-0-1-0
CIE	50	**SEE**	50

Prerequisites: None

UNIT 1:

Introduction, Modeling Concepts, Class Modeling: What is Object Orientation? What is OO development? OO themes, Modeling; abstraction; the three models, Object and class concepts; Link and associations concepts; Generalization and inheritance; A sample class model. **(09 Hrs.)**

UNIT 2:

Advanced Class Modeling: Advanced object and class concepts; Association ends; N-ary associations; Aggregation; Abstract classes, Multiple inheritance; Metadata; Reification; Constraints; Derived data; Packages.

State Modeling: Events, States, Transitions and Conditions, State diagrams, State diagram behavior. **(10 Hrs.)**

UNIT 3:

Interaction Modeling System Conception: Interaction Modeling: Use case models, Sequence models, Activity models, Use case relationships; Procedural sequence models, Special constructs for activity models, System Conception: Devising a system concept, elaborating a concept; preparing a problem statement.

Domain Analysis: Overview of analysis, Domain class model, Domain state model, Domain interaction model, Iterating the analysis. **(10 Hrs.)**

UNIT 4:

Application Analysis: Application Analysis: Application interaction model, Application class model, Application state model, adding operations, Class Design: Overview of class design, Bridging the gap, Realizing use cases, Designing algorithms, Recurring downwards, Refactoring, Design optimization, Reification of behavior, Adjustment of inheritance, Organizing a class design, ATM example. **(10 Hrs.)**

UNIT 5:

Design Patterns: Introduction: What is a design pattern, describing design patterns, how design patterns solve design problems, how to select a design pattern, how to use a design pattern, Creational Patterns – Abstract Factory, Prototype, Structural Patterns – Facade, Proxy, Behavioral Pattern – Command, Observer. **(09 Hrs.)**

List of Lab Programs – Integrated with Programming with OOMD Theory:

PART A:

Software Design using UML for the following scenarios:

1. College Management system.
2. Bank ATM.
3. E-library online public access catalogue.
4. Credit card processing system.
5. Online shopping.
6. Ticket vending machine

PART B:

Students are expected to practice the three types of design patterns namely creational, structural and behavioural pattern.

Note:

1. In I Lab Internal Test students are expected to design any one of the scenario from PART A chosen through random selection method.
2. In II Lab Internal Test students are expected to demonstrate any one design pattern with a real time scenario.(As mentioned in PART B)

Text Books:

Sl. No.	Content
1.	Michael Blaha, James Rumbaugh, "Object-Oriented Modeling and Design with UML", 2nd Edition, Pearson Education, Prentice Hall of India, 2005. Chapters 1 to 5, 7 to 13, 15 to 17 and 23
2.	E. Gamma, R. Helm, R. Johnson, J. Vlissides, "Design Patterns- Elements of Reusable Object- Oriented Software", Pearson, 1995.

Reference Books:

Sl. No.	Content
1.	Grady Booch et al, "Object-Oriented Analysis and Design with Applications", 3rd Edition, Pearson Education, 2007.
2.	Mike O'Docherty, "Object Oriented Analysis and Design", John Wiley and Sons Ltd.
3.	Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal, "Pattern-Oriented Software Architecture", Wiley Publications.

E- Books and Online Course Materials:

Sl. No.	Content
1.	Grady Booch Rational Santa Clara, "Object Oriented Analysis and Design with applications", 2nd Edition, https://dbgyan.files.wordpress.com/2013/02/grady-booch-object-oriented-analysis-and-design-with-applications-2nd-edition.pdf
2.	Buschmann, Henney, Schmidt, "Pattern- Oriented Software Architecture", http://www.ingenieria.unal.edu.co/ACSCI/sistemasycomputacion/docs/SWEBOK/Pattern-Oriented%20Software%20Architecture_%20-%20Frank%20Buschmann.pdf

Online Courses and Video Lectures:

Sl. No.	Content
1.	Prof. Partha Pratim Das, Object-Oriented Analysis and Design http://www.nptel.ac.in/courses/106105153/ Course Available from: 07-SEPTEMBER-2016 Course Co-ordinated by: IIT KHARAGPUR

Course Outcomes:

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

CO1	Describe Object Oriented Modeling and Design Techniques.
CO2	Use appropriate design pattern to solve the real world scenarios.
CO3	Design Use Case, Class, Sequence, Activity and State Diagram using UML.
CO4	Analyze the domain model and application model.

BMS COLLEGE OF ENGINEERING BENGALURU-19
(Autonomous College under VTU)
Department of Computer Applications
SEMESTER – III

COURSE TITLE	Data Science	**Credits**	4
COURSE CODE	16MCA3DCDA	**L-T-P-S**	3-0-0-1
CIE	50	**SEE**	50

Prerequisites: Mathematical Foundation to Computer Applications-II

UNIT 1:

Introduction to Data Science: The data science process. Introduction to Data mining: Why Data Mining? What Is Data Mining? What Kinds of Data Can Be Mined? What Kinds of Patterns Can Be Mined? Which Technologies Are Used? Which Kinds of Applications Are Targeted? Major Issues in Data Mining, Applications of Data mining. Getting to Know Your Data: Data Visualization, Measuring Data Similarity and Dissimilarity. **(07 Hrs.)**

UNIT 2:

Data Preprocessing: An Overview, Data Cleaning, Data Reduction - Overview of Data Reduction Strategies, PCA, Attribute Subset Selection, Histograms, Clustering, Sampling; Data Transformation and Data Discretization - Data Transformation by Normalization, Discretization by Binning, Discretization by Histogram Analysis, Discretization by Cluster, Decision Tree, and Correlation Analyses. **(07 Hrs.)**

UNIT 3:

Mining Frequent Patterns, Associations, and Correlations: Basic Concepts, Frequent Item set Mining Methods, Which Patterns Are Interesting? Pattern Evaluation Methods, Mining Rare Patterns and Negative Patterns. **(07 Hrs.)**

UNIT 4:

Classification: Basic Concepts: Basic Concepts, Decision Tree Induction, Bayes Classification Methods, k-Nearest Neighbour method.
Model Evaluation and Selection: Metrics for Evaluating Classifier Performance, Cross-validation, Bootstrap.
Techniques to Improve Classification Accuracy: Ensemble Methods, Bagging, Boosting, AdaBoost, Random Forests, Class-Imbalanced data. **(08 Hrs.)**

UNIT 5:

Cluster Analysis: Basic Concepts and Methods: Cluster Analysis, Partitioning based methods: k-Means; Hierarchical Methods: Agglomerative versus Divisive Hierarchical Clustering, Density-Based Methods: DBSCAN, Grid based methods: STING, Outlier Detection: Outliers and Outlier Analysis, Overview of Outlier Detection Methods.
(07 Hrs.)

Sample Self-study component: Simple case-studies to demonstrate application of data mining techniques using R or Python. Also a course completion from any MOOC platform like Coursera, EdX, Khan academy, etc., can be considered.

Text Books:

Sl. No.	Content
1.	Nina Zumel, and John Mount, "Practical Data Science with R", Manning Publications Co., NY, 2014, URL: https://www.manning.com/books/practical-data-science-with-r
2.	Jiawei Han and Micheline Kamber, "Data Mining: Concepts and Techniques", Third Edition, (The Morgan Kaufmann Series in Data Management Systems), 2012.

Reference Books:

Sl. No.	Content
1.	Pang-Ning Tan, Michael Steinbach, Vipin Kumar, "Introduction to Data Mining", Pearson education 2016.
2.	K.P. Soman, Shyam Diwakar, and V. Ajay, "Insight into Data mining: Theory and Practice", Prentice Hall of India Ltd, New Delhi, 2009.
3.	Ian H. Witten, Eibe Frank, Mark A. Hall, "Data Mining: Practical Machine Learning Tools and Techniques", Elsevier, 2011.
4.	Richard J. Roiger and Michael W. Geatz, "Data mining: A tutorial based primer", Pearson Education Ltd, 2003.

Online Courses and E- Books:

Sl. No.	Content
1.	Yanchang Zhao, R and Data Mining: Examples and Case Studies, http://www.RDataMining.com, 2015
2.	Zico Kolter, Carnegie Mellon University, Practical Data Science, http://www.datasciencecourse.org/
3.	Nandan Sudarsanam, IITM, Introduction to Data analytics, http://nptel.ac.in/courses/110106064/1
4.	Data mining courses, https://www.coursera.org/specializations/data-mining

Course Outcomes:

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

CO1	Explain the concepts related to data science.
CO2	Apply Pre-processing techniques for the data related to a scenario.
CO3	Devise data analytic models for a scenario.
CO4	Interpret the data to draw conclusions related to a scenario under study.

BMS COLLEGE OF ENGINEERING BENGALURU-19
(Autonomous College under VTU)
Department of Computer Applications
SEMESTER –III

COURSE TITLE	Agile Software Development	**Credits**	5
COURSE CODE	16MCA3DEAS	**L-T-P-S**	4-0-0-1
CIE	50	**SEE**	50

Prerequisites: 16MCA2DCSE – Software Engineering

UNIT 1:

Introducing Agile: WHAT IS AGILE? - The history of Agile, The Agile Manifesto, Agile Principles, THE FOUNDATIONS OF AGILE- The Agile mindset, Delivery environments and Agile suitability, The lifecycle of product development, The 'Iron Triangle', Working with uncertainty and volatility, Empirical and defined processes, AGILE AND THE BUSINESS- The economic case for Agile, Business culture and Agile. **(08 Hrs.)**

UNIT 2:

Agile Roles & Frameworks: A generic agile framework, Generic agile process -Agile operating model. Common agile roles- The customer, The team, The Agile lead, The stakeholders, Common agile techniques-Stories and backlog refinement, Agile estimation, Agile planning, Agile testing, Common agile practices-Short feedback loops, Face-to-face communication, Daily stand-ups, Show and tells, Retrospectives, Emergent documentation, Visual boards, Sustainable pace, Focus on quality, Major Agile technical practices, Agile frameworks- Major agile frameworks- eXtreme programming (XP), Scrum, Dynamic systems development method (DSDM),Kanban. **(12 Hrs.)**

UNIT 3:

Scrum Framework: Introduction- What Is Scrum? Scrum Origins, Why Scrum? Scrum Framework- Overview, Scrum Roles, Product Owner, Scrum Master, Development Team, Scrum Activities and Artifacts, Product Backlog, Sprints, Sprint Planning, Sprint Execution, Daily Scrum, Done, Sprint Review, Sprint Retrospective, Sprints – Timeboxed, Establishes a WIP Limit, Forces Prioritization, Short Duration, Ease of Planning, Fast Feedback, Improved Return on Investment, No Goal-Altering Changes, What Is a Sprint Goal?, Mutual Commitment, Change versus Clarification, What Is the Definition of Done?, Definition of Done Can Evolve Over Time, Definition of Done versus Acceptance Criteria, Done versus Done-Done. **(12 Hrs.)**

UNIT 4:

User Stories: Requirements and User Stories -Using Conversations, Progressive Refinement, What Are User Stories? Card, Conversation, Confirmation, Level of Detail, INVEST in Good Stories, Independent, Negotiable, Valuable, Estimatable, Sized Appropriately (Small), Testable, Product Backlog- What Is Grooming? Who Does the Grooming? When Does Grooming Take Place? Definition of Ready. **(08 Hrs.)**

UNIT 5:

Estimation Techniques: Estimation and Velocity- What and When We Estimate, Portfolio Backlog Item Estimates, Product Backlog Estimates, Task Estimates, PBI Estimation Concepts, Estimate as a Team, Estimates Are Not Commitments, Story Points, Ideal Days, Planning Poker, Estimation Scale
What Is Velocity? Calculate a Velocity Range, Forecasting Velocity, Technical Debt- Consequences of Technical Debt, Unpredictable Tipping Point, Increased Time to Delivery, Significant Number of Defects. **(08 Hrs.)**

Sample Self – Study: Seminar/ Presentation of Case Studies/Students can form a group and develop software using Scrum Framework e.g. Exam Management/Real time project.

Text Books:

Sl. No.	Content
1.	Peter Measey, "Agile Foundations - Principles, practices and frameworks", BCS Learning & Development Limited, 2015.
2.	Rubin, Kenneth S, "Essential Scrum: A Practical Guide to the Most Popular Agile Process", Addison-Wesley Signature Series (Cohn)). Pearson Education,2012.

Reference Books:

Sl. No.	Content
1.	Robert C. Martin Publisher, "Agile Software Development, Principles, Patterns and Practices", Prentice Hall.
2.	Ken Schawber, Mike Beedle, "Agile Software Development with Scrum", Addison Pearson.
3.	Jonathan Rasmusson, Publisher, "The Agile Samurai, How Agile Masters Deliver Great Software", SPD.
	Mike Cohn, "User Stories Applied: For Agile Software", (Addison-Wesley Signature Series (Cohn)). Pearson Education.

Course Outcomes:

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

CO1	Describe several agile methods and driving forces for taking an Agile approach to software development
CO2	Adopt best practices to successfully manage Agile projects.
CO3	Develop and monitor project backlogs, measure and monitor velocity of
CO4	Perform tailored agile processes that best fit the technical and market demands of a modern software project.

BMS COLLEGE OF ENGINEERING BENGALURU-19
(Autonomous College under VTU)
Department of Computer Applications
SEMESTER –III

COURSE TITLE	Operations Research	**Credits**	5
COURSE CODE	16MCA3DEOR	**L-T-P-S**	4-0-0-1
CIE	50	**SEE**	50

Prerequisites: None

UNIT 1:

Introduction and Overview of the OR Modeling Approach: The origin of OR, the nature of OR, the impact of OR, defining the problem and gathering data, Formulating a mathematical model, deriving solutions from the model, testing the model, preparing to apply the model, implementation.

Introduction to Linear Programming: Formulation of linear programming problem (LPP), examples, Graphical solution, the LP Model, Special cases of Graphical method, assumptions of Linear Programming (LP), additional example. **(08 Hrs.)**

UNIT 2:

Solving LPP - the Simplex Method: The essence of the simplex method, setting up the simplex method, algebra of the simplex method, the simplex method in tabular form, special cases in the simplex method, tie breaking in the simplex method, adopting to other model forms (Two Phase method, Big-M method), post optimality analysis. **(12 Hrs.)**

UNIT 3:

Transportation and Assignment Problems: The transportation problem, a stream line simplex method for the transportation problem, the assignment problem, a special algorithm for the assignment problem. **(10 Hrs.)**

UNIT 4:

Project Management: PERT and CPM: Basic difference between PERT and CPM, Phases of Project management, CPM, PERT. **(10 Hrs.)**

UNIT 5:

Game Theory: The formulation of two persons, zero sum games, solving simple games-a prototype example, games with mixed strategies, graphical solution procedure, solving by linear programming. **(08 Hrs.)**

Sample Self – Study: Case Study on Operations Research related area

Text Books:

Sl. No.	Content
1.	Frederick S.Hillier & Gerald J.Lieberman, "Introduction to Operations Research", 9th Edition, Tata McGraw Hill, 2014.
2.	J K Sharma, "Operations Research theory and applications", 5th Edition, McMillan Publications, 2013.

Reference Books:

Sl. No.	Content
1.	Wayne L. Winston, "Operations Research Applications and Algorithms", 4th Edition, Thomson Course Technology, 2003.
2.	Hamdy A. Taha, "Operations Research- An Introduction", 9th Edition, Pearson Publications, 2013.

Course Outcomes:

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

CO1	Formulate a real-world problem as a mathematical programming model.
CO2	Understand the basic concepts of different models and their applications.
CO3	Apply the models used in operations research to solve real world problem.
CO4	Analyze various models and select the best to solve a specific problem.

BMS COLLEGE OF ENGINEERING BENGALURU-19

(Autonomous College under VTU)

Department of Computer Applications

SEMESTER –III

COURSE TITLE	User Interface and User Experience	**Credits**	5
COURSE CODE	16MCA3DEUX	**L-T-P-S**	4-0-0-1
CIE	50	**SEE**	50

Prerequisites: None

UNIT 1:

What Users Do: A Means to an End, the Basics of User Research, Users' Motivation to Learn, The Patterns – Safe Exploration, Instant Gratification, Satisficing, Changes in Midstream, Deferred Choices, Incremental Construction, Habituation, Micro breaks, Spatial Memory, Prospective Memory, Streamlined Repetition, Keyboard Only, Other People's Advice, Personal Recommendations.

Information Architecture and Application Structure: The Big Picture, The Patterns – Feature, Search and browse, News Stream, Picture Manager, Dashboard, Canvas Plus Palette, Wizard, Setting Editor, Alternative Views, Many Workspaces, Multi-Level Help. **(10 Hrs.)**

UNIT 2:

Making it Look Good: Visual Style and Aesthetics: Same content, Different styles, The Basics of Visual Design, What This Means for Desktop Applications, The Patterns: Deep Background, Few Hues, Many Values, Corner Treatments, Borders That Echo Fonts, Hairlines, Contrasting Font Weights, Skins and Themes. **(08 Hrs.)**

UNIT 3:

Design and UX: Users Vs Life Cycles, Visual Design, Web standards, Potential Barriers to sustainable UX, Designing for Emerging Technologies: Design for Disruption, Eight Design Tenets for Emerging Technology, Changing Design and Designing Change, Fashion with Function: Designing for wearable devices, The next big wave in Technology, The wearable market segments, Wearables are not able, UX (and Human) Factors to consider. **(10 Hrs.)**

UNIT 4:

An Ecosystem of connected device: The concept of an Ecosystem, The 3Cs Frame work: Consistent, Continuous and Complementary, Single Device Design is History, It's an Eco system, The Consistent Design Approach: What is consistent Design, Consistency in Minimal Interface, Progressive Disclosure in Consistent Design, Beyond Device Accessibility, Devices are means not an end, The continuous Design Approach: What is Continuous Design? Single Activity flow, Sequenced Activity Flow. **(10 Hrs.)**

UNIT 5:

The Complementary Design Approach: What is Complementary Design? Collaboration: Must-Have, Collaboration: Nice to have, Control: Nice to Have, Fascinating Use Cases: What do they mean for my work? Integrated Design Approaches: 3 Cs as building blocks: Beyond the Core Devices: The Internet of Things, The Internet of Things already there? Expanding 3Cs, Multi Device Analytics: User Data is User Feedback, Multi Device Analytics and Additional Analytics Consideration. **(10 Hrs.)**

Sample Self-study: Domain specific interfaces/greener products/wearable devices/design for heterogeneous users.

Text Books:

Sl. No.	Content
1.	Jenifer Tidwell, "Designing Interfaces", 2nd Edition, Oreilly, 2015.
2.	Jonathan Follet, "Designing for Emerging Technologies- UX for Genomics, Robotics and The Internet of Things", 1st Edition, Oreilly, 2014.
3.	Michal Levin, "Designing Multi-Device Experiences", 1st Edition, Oreilly, 2014.
4.	Tim Frick, "Designing for Sustainability", 1st Edition, Oreilly 2016.

Reference Books:

Sl. No.	Content
1.	Ben Shneiderman, Plaisant, Cohen, "Jacobs: Designing the User Interface", 5th Edition, Pearson Education, 2010
2.	Unger and Chandler, "A Project Guide to UX Design", 2nd Edition, New Riders, 2012

Course Outcomes:

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

CO1	Explain the various aspects involved in designing an user interface
CO2	Demonstrate his/her work through presentations and report
CO3	Design an User interface considering the user experience for a given scenario
CO4	Collaborate in a team as a leader or a member

BMS COLLEGE OF ENGINEERING BENGALURU-19
(Autonomous College under VTU)
Department of Computer Applications
SEMESTER –III

COURSE TITLE	UNIX system programming	**Credits**	5
COURSE CODE	16MCA3DEUP	**L-T-P-S**	4-0-0-1
CIE	50	**SEE**	50

Prerequisites: None

UNIT 1:

Introduction: The POSIX Standards, The POSIX.1 FIPS Standard, The X/Open Standards. UNIX and POSIX APIs: The POSIX APIs, The UNIX and POSIX Development Environment, API Common Characteristics. **(09 Hrs.)**

UNIT 2:

File Types, The UNIX and POSIX File System, Application Program Interface to Files, UNIX Kernel Support for Files, Directory Files, Hard and Symbolic Links.
General File APIs, File and Record Locking, Directory File APIs, Device File APIs, FIFO File APIs, Symbolic Link File APIs. **(10 Hrs.)**

UNIT 3:

UNIX Processes: Introduction, main function, Process Termination, Command-Line Arguments, Environment List, Memory Layout of a C Program, Shared Libraries, Memory Allocation, Environment Variables, setjmp and longjmp. Functions, getrlimit, setrlimit Functions. **(10 Hrs.)**

UNIT 4:

Introduction, Process Identifiers, Fork, Vfork, Exit, Wait, Waitpid, waited, wait3, wait4, Functions, Race Conditions, exec Functions, Changing User IDs and Group IDs, Interpreter Files, System Function, Process Accounting, User Identification, Process Times. **(10 Hrs.)**

UNIT 5:

Network IPC: Sockets: Introduction, Socket Descriptors, Addressing, Connection establishment, Data transfer, Socket options, Out-of-band data, Nonblocking and asynchronous I/O. **(09 Hrs.)**

Sample Self-Study: Design Programs using Unix System API’s.

Text Books:

Sl. No.	Content
1.	Terrence Chan, “Unix System Programming Using C++”, Prentice-Hall of India /Pearson Education, 2008.
2.	W.Richard Stevens, Stephen A. Rago, “Advanced Programming in the UNIX Environment”, 2nd Edition, Pearson Education / Prentice-Hall of India, 2013.

Reference Books:

Sl. No.	Content
1.	Your UNIX-The Ultimate Guide, Sumitabha Das, Tata McGraw Hill, 2006
2.	W. Richard Stevens: UNIX networking Programming, Prentice-Hall of India, 1998.

Course Outcomes:

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

CO1	Explain the concepts of Unix System Programming.
CO2	Write programs using various Unix System programming API’s.
CO3	Formulate Unix commands using Unix System programming API’s.