5TH SEMESTER BCA			
	SUB CODE		SUB NAME
MAJOR	CORE-I	PAPER-11	Software Engineering
	CORE-I	PAPER-12	An Intro^{n} to AI/ An Intro^{n} to DS
	CORE-I	PAPER-13	Programming in JAVA
MINOR	CORE-II	PAPER-3	
SEC	PAPER-2		
VAC	PAPER-3		

Core XI

Software Engineering

Course Outcomes:

- To understand importance of Software engineering.
- To understand different software development models
- To understand various issues involved in a software development project

Learning Outcomes:

Upon completion of this course, students will be able to:

- Understand various software development lifecycle models
- Know the complexities involved in software development projects & how to deal with them
- Understand the software design process starting from requirement analysis
- Learn about software documentation, software testing and maintenance

Unit I:

Introduction: Evolution of Software to an Engineering Discipline, Software Development Projects, Exploratory Style of Software Development, Emergence of Software Engineering, Changes in Software Development Practices, Computer Systems Engineering. Software Lifecycle Models: Waterfall Model and its Extensions, Rapid Application Development (RAD), Agile Development Models, Spiral Model.

Unit II:

Software Project Management: Software Project Management Complexities, Responsibilities of a Software Project Manager, Project Planning, Metrics for Project Size Estimation, Project Estimation Techniques, Empirical Estimation Techniques, COCOMO, Halstead's Software Science, Staffing Level Estimation, Scheduling, Organization and Team Structures, Staffing, Risk Management, Software Configuration Management.

Unit III:

Requirement Analysis and Specification: Requirements Gathering and Analysis, Software Requirement Specifications, Formal System Specification Axiomatic Specification, Algebraic Specification, Executable Specification and 4GL.

Software Design: Design Process, Characterize a Good Software Design, Cohesion and Coupling, Layered Arrangements of Modules, Approaches to Software Design (Function Oriented & Object-Oriented).

Unit IV:

Coding and Testing: Coding: Code Review, Software Documentation, Testing, Unit Testing, Black Box and White Box Testing, Debugging, Program Analysis Tools, Integration Testing, System Testing, Software Maintenance.

Text Books:

- ✓ *Software Engineering– Ian Sommerville, 10/Ed, Pearson*
- ✓ *Fundamental of Software Engineering, Rajib Mall, Fifth Edition, PHI Publication, India.*

Reference Books:

- ✓ *Software Engineering Concepts and Practice – Ugrasen Suman, Cengage Learning India Pvt, Ltd.*
- *Software Engineering, R Khurana, Vikash Pubs*.

Core XII

(A) Introduction to Artificial Intelligence

(Students can choose any one course from this group)

Course Outcomes:

- To learn the basic concepts of AI.
- To understand AI problem-solving approaches

Learning Outcomes:

Upon completion of this course, students will be able to:

- Understand state space search as an approach to AI problem solving
- Understand various Knowledge Representation techniques
- Learn the complexity involved in NLP & role of learning in AI problem-solving
- Understand the importance of Expert systems and the use of AI programming languages.

Unit I:

Introduction to AI, Scope of AI, Characteristics of AI problems, Turing test, Concept of Intelligent agents, Approaches to AI problem-solving, State space search, production system, Uninformed search: Breadth-First, Depth-First, Iterative deepening, bidirectional and beam search.

Unit II:

Informed/Heuristic search: Generate-and-Test, Hill climbing, Best-first search, A^* algorithm, Problem reduction, AO^*, Constraint satisfaction, Solution of CSP using search, Means-End analysis.

Unit III:

Knowledge Representation: Propositional logic and Predicate logic along with their resolution principles, Unification algorithm, forward and backward chaining and conflict resolution, Semantic nets, Frames, Conceptual dependencies, Scripts.
Reasoning under uncertainty: Bayesian Belief networks, Dempster Shafer theory

Unit IV:

Natural language processing: Introduction, Levels of knowledge in language understanding, , Phases of Natural language understanding, top-down and bottom-up parsing, transition networks.
Expert Systems: Introduction, Architecture, Expert system development cycle, Examples of ES: Mycin and Dendral.

Text Books:

- ✓ *Artificial Intelligence by Rajiv Chopra, S. Chand Pubs.*
- ✓ *Artificial Intelligence by E. A. Rich and Kelvin Knight, TMH*

Reference Books:

- ✓ *Introduction to AI and Expert Systems- D.W. Patterson, PHI*
- ✓ *Principles of AI and Expert systems development, D. W. Rolston (McGraw Hill)*

(B) Introduction to Data Science

Course Objectives:

- To understand emerging issues related to various fields of data science.
- To understand the underlying principles of data science, exploring data analysis.
- To learn the basics of R Programming.

Learning Outcomes:

Upon completion of this course, students will be able to:

- Appreciate the importance of data science & learn the use different data analysis tools
- Learn R Programming
- Understand the techniques for data cleaning
- Learn the use of various data analysis and visualization tools

Unit I:

Data Scientist's Tool Box: Turning data into actionable knowledge, introduction to the tools that are used in building data analysis software: version control, markdown, git,GitHub, R, and RStudio.

Unit II:

R Programming Basics: Overview of R, R data types and objects, reading and writing data, Control structures, functions, scope rules, dates and times, Loop functions, debugging tools, Simulation, code profiling.

Unit III:

Getting and Cleaning Data: Obtaining data from the web, from APIs, from databases and other sources in various formats, basics of data cleaning and making data "tidy".

Unit IV:

Exploratory Data Analysis: Essential exploratory techniques for summarizing data, applied before formal modeling commences, eliminating or sharpening potential hypotheses about the world that can be addressed by the data, common multivariate statistical techniques used to visualize high-dimensional data.

Text Book:

- ✓ *Rachel Schutt, Cathy O'Neil, "Doing Data Science: Straight Talk from the Front line" Schroff /O'Reilly, 2013.*

Reference Books:

- ✓ *Foster Provost, Tom Fawcett, "Data Science for Business" What You Need to KnowAbout Data Mining and Data-Analytic Thinking by O'Reilly, 2013.*
- ✓ *John W. Foreman, "Data Smart: Using data Science to Transform Information into Insight"by John Wiley & Sons, 2013.*
- ✓ *Eric Seigel, "Predictive Analytics: The Power to Predict who Will Click, Buy, Lie, or Die",1st Edition, by Wiley, 2013.*

BCA 5.4B Lab: Introduction to Data Science

1. Study of basic Syntaxes in R
2. Implementation of vector data objects operations
3. Implementation of matrix, array and factors and perform variance analog in R
4. Implementation and use of data frames in R
5. Create Sample (Dummy) Data in R and perform data manipulation with R
6. Study and implementation of various control structures in R
7. Data Manipulation with dplyr package
8. Data Manipulation with data.table package
9. Study and implementation of Data Visualization with ggplot2
10. Study and implementation data transpose operations in R

Major XIII

Programming in Java

Course Outcomes:

- To learn Java for writing object-oriented programs
- To understand the use of different Java programming constructs
- To learn exception handling in Java and use of threads.

Learning Outcomes:

Upon completion of this course, students will be able to:

- Learn the basics of Java programming
- Create classes/objects and implement different forms of inheritance
- Use arrays and files in Java
- Learn about exception handling

Unit I:

Introduction to Java: Java History, Architecture and Features, Understanding the semantic and syntax differences between C++ and Java, Compiling and Executing a Java Program, Variables, Constants, Keywords (super, this, final, abstract, static, extends, implements, interface) , Data Types, Wrapper class, Operators (Arithmetic, Logical and Bitwise) and Expressions, Comments, Doing Basic Program Output, Decision Making Constructs (conditional statements and loops) and Nesting, Java Methods (Defining, Scope, Passing and Returning Arguments, Type Conversion and Type and Checking, Built-in Java Class Methods). Input through keyboard using Command line Argument, the Scanner class, BufferedReader class.

Unit II:

Object-Oriented Programming Overview: Principles of Object-Oriented Programming, Defining & Using Classes, Class Variables & Methods, Objects, Object reference, Objects as parameters, final classes, Garbage Collection. Constructor- types of constructors, this keyword, super keyword. Method overloading and Constructor overloading. Aggregation vs Inheritance, Inheritance: extends vs implements, types of Inheritance, Interface, Up-Casting, Down-Casting, Auto-Boxing, Enumerations, Polymorphism, Method Overriding and restrictions. Package: Pre-defined packages and Custom packages.

Unit III:

Arrays: Creating & Using Arrays (1D, 2D, 3D and Jagged Array), Array of Object, Referencing Arrays Dynamically. Strings and I/O: Java Strings: The Java String class, Creating & Using String Objects, Manipulating Strings, String Immutability& Equality, Passing Strings To & From Methods, StringBuffer Classes and StringBuilder Classes. IO package: Understanding StreamsFile class and its methods, Creating, Reading, Writing using classes: Byte and Character streams, FileOutputStream, FileInputStream, FileWriter, FileReader, InputStreamReader, PrintStream, PrintWriter. Compressing and Uncompressing File.

Unit IV:

Exception Handling, Threading, Networking and Database Connectivity: Exception types, uncaught exceptions, throw, built-in exceptions, Creating your own exceptions; Multi-

threading: The Thread class and Runnable interface, creating single and multiple threads, Thread prioritization, synchronization and communication, suspending/resuming threads. Using java.net package, Overview of TCP/IP and Datagram programming. Accessing and manipulating databases using JDBC.

Text Book:

- ✓ *E. Balagurusamy, "Programming with Java", TMH, 4/Ed*

Reference Book:

- *Herbert Schildt, "The Complete Reference to Java", TMH, 10/Ed.*

BCA 6.1 Lab: Programming in Java

1. To find the sum of any number of integers entered as command line arguments.
2. To find the factorial of a given number.
3. To convert a decimal to binary number.
4. To check if a number is prime or not, by taking the number as input from the keyboard.
5. To find the sum of any number of integers interactively, i.e., entering every number from the keyboard, whereas the total number of integers is given as a command line argument.
6. Write a program that show working of different functions of String and StringBufferclasss like setCharAt( ), setLength( ), append( ), insert( ), concat( )and equals( ).
7. Write a program to create a – "distance" class with methods where distance is computed in terms of feet and inches, how to create objects of a class and to see the use of this pointer
8. Modify the – "distance" class by creating constructor for assigning values (feetandinches) to the distance object. Create another object and assign second object as reference variable to another object reference variable. Further create a third object which is a clone of the first object.
9. Write a program to show that during function overloading, if no matching argument is found, then Java will apply automatic type conversions (from lower to higher data type).
10. Write a program to show the difference between public and private access specifiers. The program should also show that primitive data types are passed by value and objects are passed by reference and to learn use of final keyword.
11. Write a program to show the use of static functions and to pass variable length arguments in a function.
12. Write a program to demonstrate the concept of boxing and unboxing.
13. Create a multi-file program where in one file a string message is taken as input from the user and the function to display the message on the screen is given in another file (make use of Scanner package in this program).
14. Write a program to create a multilevel package and also creates a reusable class to generate Fibonacci series, where the function to generate Fibonacci series is given in a different file belonging to the same package.
15. Write a program that creates illustrates different levels of protection in classes/subclasses belonging to same package or different packages

16. Write a program – "DivideByZero" that takes two numbers a and b as input, computes a/b, and invokes Arithmetic Exception to generate a message when the denominator is zero.
17. Write a program to show the use of nested try statements that emphasizes the sequence of checking for catch handler statements.
18. Write a program to create your own exception types to handle situation specific to your application (Hint: Define a subclass of Exception which itself is a subclass of Throwable).
19. Write a program to demonstrate priorities among multiple threads.
20. Write a program to demonstrate different mouse handling events like mouseClicked(), mouseEntered(), mouseExited(), mousePressed(), mouseReleased() & mouseDragged().
21. Write a program to demonstrate different keyboard handling events.