**MODEL PAPER – 1**
**Bachelor of Computer Application (BCA)– Semester 3**
**Paper: Object Oriented Programming using Java**
**Time: 3 Hours | Max Marks**: 70

## ◇ SECTION – A

**Answer any SIX questions (6 × 5 = 30 Marks)**

**1.**

a) Define Object-Oriented Programming and explain its basic concepts.

b) Explain features of Java.

c) Describe Java program structure.

d) Define tokens in Java.

e) Explain data types in Java.

f) Define operators and explain types of operators.

g) What is class and object?

h) Define exception handling.

## ◇ SECTION – B

**Answer ALL questions (4 × 10 = 40 Marks)**

**2.**(a) Explain benefits and applications of Object-Oriented Programming.
**OR**
(b) Explain Java environment and JVM in detail.

**3.**(a) Explain constants, variables, and data types with examples.
**OR**
(b) Explain operators and expressions including precedence and associativity.

**4.**(a) Explain classes, objects, and constructors with suitable examples.
**OR**
(b) Explain inheritance and its types with examples.

**5.**(a) Explain arrays and strings in Java.
**OR**
(b) Explain exception handling using try, catch, throw, throws, and finally.

**MODEL PAPER – 2**
**Bachelor of Computer Application (BCA)– Semester 3**
**Paper: Object Oriented Programming using Java**
**Time: 3 Hours | Max Marks**: 70

## ◇ SECTION – A

**Answer any SIX questions (6 × 5 = 30 Marks)**

**1.**

a) Explain evolution of Java and its applications.

b) Define command line arguments with example.

c) Explain symbolic constants in Java.

d) Define type casting with example.

e) Explain relational and logical operators.

f) Define looping statements in Java.

g) Explain method overloading.

h) Define package in Java.

## ◇ SECTION – B

**Answer ALL questions (4 × 10 = 40 Marks)**

**2.**(a) Explain Java features and compare Java with C++ language.
**OR**
(b) Explain Java tokens and statements with examples.

**3.**(a) Explain decision making and branching statements with examples.
**OR**
(b) Explain looping statements and jump statements in Java.

**4.**(a) Explain constructors and method overloading with examples.
**OR**
(b) Explain inheritance and polymorphism in Java.

**5.**(a) Explain arrays and vectors with suitable examples.
**OR**
(b) Explain packages and creation of user-defined packages.

**MODEL PAPER – 3**
**Bachelor of Computer Application (BCA)– Semester 3**
**Paper: Object Oriented Programming using Java**
**Time: 3 Hours | Max Marks**: 70

## ◇ SECTION – A

**Answer any SIX questions (6 × 5 = 30 Marks)**

**1.**

a) Define Java environment and explain its components.

b) Explain comments and Math functions in Java.

c) Define arithmetic expressions in Java.

d) Explain increment and decrement operators.

e) Define overriding methods.

f) Explain strings in Java.

g) Define wrapper classes.

h) Explain try and catch blocks.

## ◇ SECTION – B

**Answer ALL questions (4 × 10 = 40 Marks)**

**2.**(a) Explain Java program execution process including JVM.
**OR**
(b) Explain constants, variables, and type conversion in Java.

**3.**(a) Explain operators and expression evaluation with examples.
**OR**
(b) Explain decision making statements and labelled loops.

**4.**(a) Explain classes, constructors, and method overriding with examples.
**OR**
(b) Explain interfaces and implementation of interfaces.

**5.**(a) Explain arrays and string handling in Java.
**OR**
(b) Explain exception handling including multiple catch and nested try.

**MODEL PAPER – 4**
**Bachelor of Computer Application (BCA)– Semester 3**
**Paper: Object Oriented Programming using Java**
**Time: 3 Hours | Max Marks**: 70

## ◇ SECTION – A

**Answer any SIX questions (6 × 5 = 30 Marks)**

**1.**

a) Define OOP and explain its features.

b) Explain Java program structure.

c) Define data types and variables.

d) Explain operators in Java.

e) Define class and object.

f) Explain inheritance.

g) Define arrays.

h) Explain exception handling.

## ◇ SECTION – B

**2.**(a) Explain Java features and JVM.
**OR**
(b) Explain tokens and statements in Java.

**3.**(a) Explain control statements and loops.
**OR**
(b) Explain operators and expressions.

**4.**(a) Explain classes, constructors, and inheritance.
**OR**
(b) Explain polymorphism and method overloading/overriding.

**5.**(a) Explain arrays, strings, and vectors.
**OR**
(b) Explain exception handling with examples.

**MODEL PAPER – 5**
**Bachelor of Computer Application (BCA)– Semester 3**
**Paper: Object Oriented Programming using Java**
**Time: 3 Hours | Max Marks**: 70

## ◇ SECTION – A

**Answer any SIX questions (6 × 5 = 30 Marks)**

**1.**

a) Define Object-Oriented Programming and explain its features.

b) Explain Java program structure with example.

c) Define data types and variables in Java.

d) Explain operators and expressions.

e) Define class and object with example.

f) Explain inheritance and its types.

g) Define arrays and explain types of arrays.

h) Explain exception handling in Java.

## ◇ SECTION – B

**Answer ALL questions (4 × 10 = 40 Marks)**

**2.**(a) Explain Java features and JVM execution process.
**OR**
(b) Explain tokens and statements in Java with examples.

**3.**(a) Explain control statements including decision making and looping with examples.
**OR**
(b) Explain operators and expression evaluation with precedence rules.

**4.**(a) Explain classes, constructors, and inheritance with suitable examples.
**OR**
(b) Explain polymorphism including method overloading and overriding.

**5.**(a) Explain arrays, strings, and vectors with suitable examples.

**OR**

(b) Explain exception handling using try, catch, throw, throws, and finally with example.

## UNIT WISE IMPORTANT QUESTIONS

### UNIT – I (OOP Basics & Java Fundamentals)

- Basic concepts of OOP ⭐⭐⭐⭐⭐
- Benefits and applications of OOP ⭐⭐⭐⭐⭐
- Features of Java ⭐⭐⭐⭐⭐
- Difference between Java, C, and C++ ⭐⭐⭐⭐
- Java environment and JVM ⭐⭐⭐⭐⭐
- Structure of Java program ⭐⭐⭐⭐⭐
- Java tokens and statements ⭐⭐⭐⭐⭐
- Command line arguments ⭐⭐⭐⭐⭐

### UNIT – II (Data Types, Operators & Control Statements)

- Constants and variables ⭐⭐⭐⭐⭐
- Data types in Java ⭐⭐⭐⭐⭐
- Type casting ⭐⭐⭐⭐⭐
- Operators (all types) ⭐⭐⭐⭐⭐
- Operator precedence and associativity ⭐⭐⭐⭐⭐
- Expression evaluation ⭐⭐⭐⭐
- Decision making statements ⭐⭐⭐⭐⭐
- Looping statements ⭐⭐⭐⭐⭐
- Jump statements ⭐⭐⭐⭐

### UNIT – III (Classes, Inheritance & Arrays)

- Classes and objects ⭐⭐⭐⭐⭐
- Constructors ⭐⭐⭐⭐⭐
- Method overloading ⭐⭐⭐⭐⭐
- Method overriding ⭐⭐⭐⭐⭐
- Inheritance (all types) ⭐⭐⭐⭐⭐
- Interfaces ⭐⭐⭐⭐⭐
- Arrays (1D & 2D) ⭐⭐⭐⭐⭐
- Strings ⭐⭐⭐⭐⭐
- Vectors ⭐⭐⭐⭐

- Wrapper classes ⭐⭐⭐⭐

## UNIT – IV (Packages & Exception Handling)

- Packages (system & user-defined) ⭐⭐⭐⭐⭐
- Creating and accessing packages ⭐⭐⭐⭐⭐
- Exception handling ⭐⭐⭐⭐⭐
- try, catch, throw, throws, finally ⭐⭐⭐⭐⭐
- Multiple catch blocks ⭐⭐⭐⭐⭐
- Nested try ⭐⭐⭐⭐
- User-defined exceptions ⭐⭐⭐⭐⭐

**MODEL PAPER – 1**
**Bachelor of Computer Application (BCA)– Semester 3**
**Paper: Design and Analysis of Algorithms**
**Time: 3 Hours | Max Marks**: 70

## ◇ SECTION – A

**Answer any SIX questions (6 × 5 = 30 Marks)**

**1.**

a) Define algorithm and explain its characteristics.

b) Explain time and space complexity.

c) Define asymptotic notations with examples.

d) Explain bubble sort algorithm.

e) Define recursion with example.

f) Explain binary search algorithm.

g) Define greedy method.

h) Explain Breadth First Search (BFS).

## ◇ SECTION – B

**Answer ALL questions (4 × 10 = 40 Marks)**

**2.**(a) Explain asymptotic notations (O, Ω, Θ) with suitable examples.
**OR**
(b) Explain analysis of sorting algorithms: bubble, selection, and insertion sort.

**3.**(a) Explain divide and conquer technique with merge sort.
**OR**
(b) Explain quick sort algorithm with complexity analysis.

**4.**(a) Explain greedy method with Prim's or Kruskal's algorithm.
**OR**
(b) Explain Dijkstra's algorithm for shortest path.

**5.**(a) Explain dynamic programming with 0/1 Knapsack problem.
**OR**
(b) Explain DFS and BFS traversal techniques.

**MODEL PAPER – 2**
**Bachelor of Computer Application (BCA)– Semester 3**
**Paper: Design and Analysis of Algorithms**
**Time: 3 Hours | Max Marks**: 70

## ◇ SECTION – A

**Answer any SIX questions (6 × 5 = 30 Marks)**

**1.**

a) Define time complexity and space complexity.

b) Explain insertion sort algorithm.

c) Define Master's theorem.

d) Explain matrix multiplication.

e) Define divide and conquer technique.

f) Explain minimum and maximum finding algorithm.

g) Define dynamic programming.

h) Explain Depth First Search (DFS).

## ◇ SECTION – B

**Answer ALL questions (4 × 10 = 40 Marks)**

**2.**(a) Explain analysis of sequential search and binary search.
**OR**
(b) Explain recursive algorithms and solve recurrence using Master's theorem.

**3.**(a) Explain divide and conquer technique with binary search and merge sort.
**OR**
(b) Explain quick sort with best and worst case analysis.

**4.**(a) Explain greedy method with knapsack problem.
**OR**
(b) Explain minimum spanning tree using Prim's or Kruskal's algorithm.

**5.**(a) Explain dynamic programming using Fibonacci or binomial coefficients.
**OR**
(b) Explain graph algorithms including BFS and DFS.

**MODEL PAPER – 3**
**Bachelor of Computer Application (BCA)– Semester 3**
**Paper: Design and Analysis of Algorithms**
**Time: 3 Hours | Max Marks**: 70

## ◇ SECTION – A

**Answer any SIX questions (6 × 5 = 30 Marks)**

**1.**

a) Define space complexity with example.

b) Explain sequential search algorithm.

c) Define best case and worst case analysis.

d) Explain lower bound for comparison-based sorting.

e) Define Floyd-Warshall algorithm.

f) Explain connected components in graphs.

g) Define backtracking technique.

h) Explain NP-Hard problems.

## ◇ SECTION – B

**Answer ALL questions (4 × 10 = 40 Marks)**

**2.**(a) Explain algorithm design and performance analysis with suitable examples.
**OR**
(b) Explain different cases of complexity analysis with examples.

**3.**(a) Explain divide and conquer technique for finding maximum and minimum elements.
**OR**
(b) Explain lower bound theory for sorting algorithms.

**4.**(a) Explain dynamic programming approach using Floyd-Warshall algorithm.
**OR**
(b) Explain graph algorithms for connected components and directed graphs.

**5.**(a) Explain backtracking method using N-Queen problem.
**OR**
(b) Explain computational complexity including P, NP, and NP-Complete problems.

**MODEL PAPER – 4**
**Bachelor of Computer Application (BCA)– Semester 3**
**Paper: Design and Analysis of Algorithms**
**Time: 3 Hours | Max Marks**: 70

## ◇ SECTION – A

**Answer any SIX questions (6 × 5 = 30 Marks)**

**1.**

a) Define algorithm design technique with example.

b) Explain evaluation of expressions in algorithm analysis.

c) Define recurrence relation with example.

d) Explain merge sort algorithm.

e) Define knapsack problem.

f) Explain Prim's algorithm.

g) Define binomial coefficients.

h) Explain Hamiltonian circuit problem.

## ◇ SECTION – B

**Answer ALL questions (4 × 10 = 40 Marks)**

**2.**(a) Explain time and space complexity with suitable examples.
**OR**
(b) Explain asymptotic notations and their applications.

**3.**(a) Explain divide and conquer technique with merge sort analysis.
**OR**
(b) Explain greedy method with knapsack problem.

**4.**(a) Explain dynamic programming using binomial coefficients.
**OR**
(b) Explain graph algorithms including BFS and DFS with examples.

**5.**(a) Explain backtracking method using sum of subsets problem.
**OR**
(b) Explain NP-Complete problems and computational intractability.

**MODEL PAPER – 4**
**Bachelor of Computer Application (BCA)– Semester 3**
**Paper: Design and Analysis of Algorithms**
**Time: 3 Hours | Max Marks**: 70

## ◇ SECTION – A

**Answer any SIX questions (6 × 5 = 30 Marks)**

**1.**

a) Define asymptotic notations and explain Big-O notation with example.

b) Explain insertion sort with time complexity.

c) Define recurrence relation and give example.

d) Explain binary search using divide and conquer.

e) Define greedy method and its characteristics.

f) Explain 0/1 Knapsack problem.

g) Define Breadth First Search (BFS).

h) Explain N-Queen problem.

## ◇ SECTION – B

**Answer ALL questions (4 × 10 = 40 Marks)**

**2.**(a) Explain asymptotic notations (O, Ω, Θ) and analyze growth of functions.
**OR**
(b) Analyze sorting algorithms (bubble, selection, insertion) with complexity comparison.

**3.**(a) Explain divide and conquer technique with merge sort and quick sort.
**OR**
(b) Solve recurrence relations using Master's theorem with examples.

**4.**(a) Explain greedy method with minimum spanning tree algorithms (Prim's and Kruskal's).
**OR**
(b) Explain shortest path problem using Dijkstra's algorithm.

**5.**(a) Explain dynamic programming using 0/1 Knapsack and Floyd-Warshall algorithm.
**OR**
(b) Explain backtracking method using N-Queen and sum of subsets problem.

## UNIT-WISE IMPORTANT QUESTIONS

### UNIT – I (Algorithm Basics & Complexity)

- Definition and characteristics of an algorithm ⭐⭐⭐⭐⭐
- Time and space complexity ⭐⭐⭐⭐⭐
- Asymptotic notations (O, Ω, Θ) ⭐⭐⭐⭐⭐
- Analysis of sequential search ⭐⭐⭐⭐⭐
- Bubble sort ⭐⭐⭐⭐⭐
- Selection sort ⭐⭐⭐⭐⭐
- Insertion sort ⭐⭐⭐⭐⭐
- Matrix multiplication ⭐⭐⭐⭐
- Recursion concept ⭐⭐⭐⭐⭐
- Analysis of recursive algorithms ⭐⭐⭐⭐⭐
- Master's theorem ⭐⭐⭐⭐⭐

### UNIT – II (Divide & Conquer & Greedy)

- Divide and Conquer technique ⭐⭐⭐⭐⭐
- Binary search ⭐⭐⭐⭐⭐
- Merge sort ⭐⭐⭐⭐⭐
- Quick sort ⭐⭐⭐⭐⭐
- Best and worst case analysis ⭐⭐⭐⭐⭐
- Strassen's matrix multiplication ⭐⭐⭐⭐
- Lower bound for sorting ⭐⭐⭐⭐⭐
- Greedy method ⭐⭐⭐⭐⭐
- Knapsack problem ⭐⭐⭐⭐⭐
- Prim's algorithm ⭐⭐⭐⭐⭐
- Kruskal's algorithm ⭐⭐⭐⭐⭐
- Dijkstra's algorithm ⭐⭐⭐⭐⭐

### UNIT – III (Dynamic Programming & Graphs)

- Dynamic programming concept ⭐⭐⭐⭐⭐
- Fibonacci using DP ⭐⭐⭐⭐⭐
- Binomial coefficients ⭐⭐⭐⭐⭐
- Floyd-Warshall algorithm ⭐⭐⭐⭐⭐
- 0/1 Knapsack problem ⭐⭐⭐⭐⭐
- Breadth First Search (BFS) ⭐⭐⭐⭐⭐
- Depth First Search (DFS) ⭐⭐⭐⭐⭐
- Connected components ⭐⭐⭐⭐
- Topological sorting ⭐⭐⭐⭐⭐

## UNIT – IV (Backtracking & NP Problems)

- Backtracking method
- N-Queen problem
- Sum of subsets problem
- Hamiltonian circuit
- Vertex cover problem
- P, NP, NP-Complete problems
- NP-Hard problems

**MODEL PAPER – 1**
**Bachelor of Computer Application (BCA)– Semester 3**
**Paper: Operating System**
**Time: 3 Hours | Max Marks**: 70

## ◇ SECTION – A

**Answer any SIX questions (6 × 5 = 30 Marks)**
**1.**

**Answer any SIX questions (6 × 5 = 30 Marks)**

a) Define operating system and explain its functions.

b) Explain types of operating systems.

c) Define system calls with examples.

d) Explain process and process states.

e) Define process control block (PCB).

f) Explain threads and their benefits.

g) Define critical section problem.

h) Explain paging technique.

## ◇ SECTION – B

**Answer ALL questions (4 × 10 = 40 Marks)**

**2.**(a) Explain operating system structure and services.
**OR**
(b) Explain batch systems, multiprogramming, and time-sharing systems.

**3.**(a) Explain process scheduling and scheduling algorithms (FCFS, SJF, RR).
**OR**
(b) Explain threads and multithreading models.

**4.**(a) Explain process synchronization and semaphores.
**OR**
(b) Explain deadlocks and methods for prevention and avoidance.

**5.**(a) Explain memory management including paging and segmentation.
**OR**
(b) Explain virtual memory and page replacement algorithms.

**MODEL PAPER – 2**
**Bachelor of Computer Application (BCA)– Semester 3**
**Paper: Operating System**
**Time: 3 Hours | Max Marks**: 70

## ◇ SECTION – A

**Answer any SIX questions (6 × 5 = 30 Marks)**

**1.**

a) Define operating system and its components.

b) Explain OS structure.

c) Define process and process states.

d) Explain scheduling algorithms.

e) Define synchronization.

f) Explain deadlock.

g) Define paging.

h) Explain virtual memory.

## ◇ SECTION – B

**2.**(a) Explain types of operating systems and their features.
**OR**
(b) Explain system calls and OS services

**3.**(a) Explain CPU scheduling algorithms (FCFS, SJF, RR).
**OR**
(b) Explain threads and multithreading.

**4.**(a) Explain synchronization techniques using semaphores.
**OR**
(b) Explain deadlock prevention and avoidance.

**5.**(a) Explain memory management including paging and segmentation.
**OR**
(b) Explain virtual memory and page replacement algorithms.

**MODEL PAPER – 3**
**Bachelor of Computer Application (BCA)– Semester 3**
**Paper: Operating System**
**Time: 3 Hours | Max Marks**: 70

## ◇ SECTION – A

**Answer any SIX questions (6 × 5 = 30 Marks)**
**1.**

**Answer any SIX questions (6 × 5 = 30 Marks)**

a) Define time-sharing system.

b) Explain operating system services.

c) Define threads and types of threads.

d) Explain round robin scheduling.

e) Define mutual exclusion.

f) Explain deadlock characterization.

g) Define virtual memory.

h) Explain demand paging.

## ◇ SECTION – B

**Answer ALL questions (4 × 10 = 40 Marks)**

**2.**(a) Explain types of operating systems with examples.
**OR**
(b) Explain system calls and system programs.

**3.**(a) Explain process states and process state transitions.
**OR**
(b) Explain scheduling algorithms with suitable examples.

**4.**(a) Explain semaphores and critical section problem.
**OR**
(b) Explain Banker's algorithm with suitable example.

**5.**(a) Explain paging and segmentation techniques.
**OR**
(b) Explain page replacement algorithms and thrashing.

**MODEL PAPER – 4**
**Bachelor of Computer Application (BCA)– Semester 3**
**Paper: Operating System**
**Time: 3 Hours | Max Marks**: 70

## ◇ SECTION – A

**Answer any SIX questions (6 × 5 = 30 Marks)**
**1.**

**Answer any SIX questions (6 × 5 = 30 Marks)**

a) Define multiprogramming and its advantages.

b) Explain operating system architecture types.

c) Define process scheduling criteria.

d) Explain multithreading and its models.

e) Define inter-process communication (IPC).

f) Explain deadlock prevention techniques.

g) Define fragmentation and compaction.

h) Explain disk structure.

## ◇ SECTION – B

**Answer ALL questions (4 × 10 = 40 Marks)**

**2.**(a) Explain evolution and generations of operating systems.
**OR**
(b) Explain operating system structure with suitable examples.

**3.**(a) Explain process management and role of PCB in detail.
**OR**
(b) Explain CPU scheduling with comparison of preemptive and non-preemptive methods.

**4.**(a) Explain synchronization problems and solutions using monitors.
**OR**
(b) Explain deadlock avoidance using Banker's algorithm with illustration.

**5.**(a) Explain memory management techniques including swapping and allocation strategies.
**OR**
(b) Explain disk scheduling algorithms and their performance.

**MODEL PAPER – 5**
**Bachelor of Computer Application (BCA)– Semester 3**
**Paper: Operating System**
**Time: 3 Hours | Max Marks**: 70

## ◇ SECTION – A

**Answer any SIX questions (6 × 5 = 30 Marks)**
**1.**

**Answer any SIX questions (6 × 5 = 30 Marks)**

a) Define real-time operating system.

b) Explain system call categories.

c) Define thread life cycle.

d) Explain shortest job first scheduling.

e) Define semaphore and its types.

f) Explain deadlock detection.

g) Define thrashing.

h) Explain page replacement strategies.

## ◇ SECTION – B

**Answer ALL questions (4 × 10 = 40 Marks)**

**2.**(a) Explain types of operating systems with suitable examples.
**OR**
(b) Explain operating system services and system call interface.

**3.**(a) Explain process scheduling algorithms with performance comparison.
**OR**
(b) Explain multithreading concepts and advantages.

**4.**(a) Explain critical section problem and solutions.
**OR**
(b) Explain deadlock detection and recovery methods.

**5.**(a) Explain virtual memory and demand paging in detail.
**OR**
(b) Explain disk scheduling algorithms with examples.

**MODEL PAPER – 6**
**Bachelor of Computer Application (BCA)– Semester 3**
**Paper: Operating System**
**Time: 3 Hours | Max Marks**: 70

## ◇ SECTION – A

**Answer any SIX questions (6 × 5 = 30 Marks)**

**1.**

a) Explain components of operating system.

b) Define distributed operating system.

c) Explain system programs.

d) Define process scheduling objectives.

e) Explain preemptive and non-preemptive scheduling.

f) Define race condition.

g) Explain Banker's algorithm.

h) Define segmentation.

## ◇ SECTION – B

**Answer ALL questions (4 × 10 = 40 Marks)**

**2.**(a) Explain operating system architecture and generations.
**OR**
(b) Explain real-time and parallel operating systems.

**3.**(a) Explain CPU scheduling algorithms with examples.
**OR**
(b) Explain performance evaluation of scheduling algorithms.

**4.**(a) Explain inter-process communication and synchronization problems.
**OR**
(b) Explain deadlock detection and recovery techniques.

**5.**
(a) Explain memory allocation techniques (MFT, MVT) with fragmentation.
**OR**
(b) Explain disk scheduling algorithms with examples.

## UNIT WISEIMPORTANT QUESTIONS

### UNIT – I (OS Basics & Structure)

- Definition and functions of Operating System
- Components and services of OS
- Types of Operating Systems
- Multiprogramming and Time-sharing
- Batch, Real-time, Distributed OS
- OS structure and architecture
- System calls
- System programs

### UNIT – II (Process Management & Scheduling)

- Process and process states
- Process Control Block (PCB)
- Threads and multithreading
- Scheduling objectives
- FCFS scheduling
- SJF scheduling
- Round Robin scheduling
- Preemptive vs Non-preemptive
- Performance evaluation

### UNIT – III (Synchronization & Deadlocks)

- Process synchronization
- Inter-process communication (IPC)
- Race condition
- Critical section problem
- Semaphores
- Monitors
- Deadlock characterization
- Deadlock prevention
- Banker's algorithm
- Deadlock detection & recovery

### UNIT – IV (Memory & I/O Management)

- Memory management basics
- Paging
- Segmentation
- Fragmentation & compaction

- Virtual memory ⭐⭐⭐⭐⭐
- Demand paging ⭐⭐⭐⭐⭐
- Page replacement algorithms ⭐⭐⭐⭐⭐
- Thrashing ⭐⭐⭐⭐⭐
- Disk structure ⭐⭐⭐⭐⭐
- Disk scheduling algorithms ⭐⭐⭐⭐⭐

**MODEL PAPER – 1**
**Bachelor of Computer Application (BCA)– Semester 3**
**Paper: Software Engineering**
**Time: 3 Hours | Max Marks**: 70

---

## ◇ SECTION – A

**Answer any SIX questions (6 × 5 = 30 Marks)**

**1.**

a) Define software and explain its evolving nature.

b) Explain layered technology in software engineering.

c) Define waterfall model with advantages and disadvantages.

d) Explain agile software development principles.

e) Define Software Requirements Specification (SRS).

f) Explain requirement elicitation and validation.

g) Define software design and its importance.

h) Explain software testing strategies.

---

## ◇ SECTION – B

**Answer ALL questions (4 × 10 = 40 Marks)**

**2.**(a) Explain process models including waterfall and incremental models.
**OR**
(b) Explain agile methods and compare agile with plan-driven development.

**3.**(a) Explain requirements engineering process with suitable examples.
**OR**
(b) Explain risk management including risk identification and RMMM plan.

**4.**(a) Explain software design concepts and architectural design.
**OR**
(b) Explain UML diagrams (class, sequence, use case) with examples.

**5.**(a) Explain testing strategies including black-box and white-box testing.
**OR**
(b) Explain software quality assurance and software reviews.

**MODEL PAPER – 2**
**Bachelor of Computer Application (BCA)– Semester 3**
**Paper: Software Engineering**
**Time: 3 Hours | Max Marks**: 70

## ◇ SECTION – A

**Answer any SIX questions (6 × 5 = 30 Marks)**

**1.**

a) Define software process and explain process framework.

b) Explain unified process model.

c) Define functional and non-functional requirements.

d) Explain requirement management.

e) Define software project planning.

f) Explain estimation techniques.

g) Define software metrics.

h) Explain software maintenance.

## ◇ SECTION – B

**2.**(a) Explain software process models and their comparison.
**OR**
(b) Explain agile development including Scrum and XP.

**3.**(a) Explain requirements engineering and SRS in detail.
**OR**
(b) Explain risk management strategies and risk projection.

**4.**(a) Explain software design process and architectural design.
**OR**
(b) Explain testing strategies and debugging techniques.

**5.**(a) Explain software quality assurance and reliability.
**OR**
(b) Explain release management and software maintenance.

**MODEL PAPER – 3**
**Bachelor of Computer Application (BCA)– Semester 3**
**Paper: Software Engineering**
**Time: 3 Hours | Max Marks**: 70

## ◇ SECTION – A

**Answer any SIX questions (6 × 5 = 30 Marks)**

**1.**

a) Define process framework and explain its activities.

b) Explain incremental and evolutionary process models.

c) Define Extreme Programming (XP).

d) Explain software requirements and their types.

e) Define risk and explain types of software risks.

f) Explain project scheduling.

g) Define software architecture.

h) Explain black-box testing.

## ◇ SECTION – B

**Answer ALL questions (4 × 10 = 40 Marks)**

**2.**(a) Explain process models including unified process with suitable examples.
**OR**
(b) Explain agile development and compare it with plan-driven development.

**3.**(a) Explain requirements engineering process including elicitation and validation.
**OR**
(b) Explain risk management including risk identification, projection, and RMMM.

**4.**(a) Explain software design concepts and design model with examples.
**OR**
(b) Explain UML diagrams including class, sequence, and collaboration diagrams.

**5.**(a) Explain testing strategies including white-box and validation testing.
**OR**
(b) Explain software metrics and quality measurement techniques.

**MODEL PAPER – 4**
**Bachelor of Computer Application (BCA)– Semester 3**
**Paper: Software Engineering**
**Time: 3 Hours | Max Marks**: 70

## ◇ SECTION – A

**Answer any SIX questions (6 × 5 = 30 Marks)**

**1.**

a) Define software engineering and explain its importance.

b) Explain layered approach in software engineering.

c) Define Scrum and explain its roles.

d) Explain non-functional requirements with examples.

e) Define risk refinement.

f) Explain estimation techniques in software engineering.

g) Define design model.

h) Explain white-box testing.

## ◇ SECTION – B

**Answer ALL questions (4 × 10 = 40 Marks)**

**2.**(a) Explain software process and process framework with examples.
**OR**
(b) Explain agile methods including Scrum and XP.

**3.**(a) Explain requirements specification and SRS document.
**OR**
(b) Explain risk management strategies including reactive and proactive approaches.

**4.**(a) Explain software design process and architectural design.
**OR**
(b) Explain UML diagrams including use case and component diagrams.

**5.**(a) Explain testing strategies including system testing and validation testing.
**OR**
(b) Explain software quality assurance and formal technical reviews.

**MODEL PAPER – 5**
**Bachelor of Computer Application (BCA)– Semester 3**
**Paper: Software Engineering**
**Time: 3 Hours | Max Marks**: 70

## ◇ SECTION – A

**Answer any SIX questions (6 × 5 = 30 Marks)**

**1.**

a) Define software process model and explain its types.

b) Explain unified process model.

c) Define agility and explain agile principles.

d) Explain requirements validation.

e) Define software pricing.

f) Explain project scheduling techniques.

g) Define software reliability.

h) Explain release planning.

## ◇ SECTION – B

**Answer ALL questions (4 × 10 = 40 Marks)**

**2.**(a) Explain software process models and compare waterfall and incremental models.
**OR**
(b) Explain agile development including Scrum and Extreme Programming (XP).

**3.**(a) Explain requirements engineering process and SRS document.
**OR**
(b) Explain risk management including risk identification, projection, and refinement.

**4.**(a) Explain software design concepts and architectural design.
**OR**
(b) Explain UML diagrams including class and sequence diagrams.

**5.**(a) Explain testing strategies including black-box and white-box testing.
**OR**
(b) Explain software quality assurance and software reliability.

**MODEL PAPER – 6**
**Bachelor of Computer Application (BCA)– Semester 3**
**Paper: Software Engineering**
**Time: 3 Hours | Max Marks**: 70

## ◇ SECTION – A

**Answer any SIX questions (6 × 5 = 30 Marks)**

**1.**

a) Define software engineering and layered technology.

b) Explain waterfall model.

c) Define functional and non-functional requirements.

d) Explain SRS document.

e) Define risk management.

f) Explain estimation techniques.

g) Define software testing.

h) Explain software maintenance.

## ◇ SECTION – B

**2.**(a) Explain process models and agile development.
**OR**
(b) Explain unified process and incremental model.

**3.**(a) Explain requirements engineering and validation.
**OR**
(b) Explain risk management and RMMM plan.

**4.**(a) Explain software design and UML diagrams.
**OR**
(b) Explain testing strategies and debugging.

**5.**(a) Explain software quality assurance and reviews.
**OR**
(b) Explain release management and maintenance.

## UNIT WISE IMPORTANT QUESTIONS

### UNIT – I (Software Process & Agile)

- Evolving role of software
- Characteristics of modern software
- Layered technology
- Software process framework
- Waterfall model
- Incremental process model
- Evolutionary models
- Unified process
- Agile principles
- Agile vs Plan-driven development
- Extreme Programming (XP)
- Scrum

### UNIT – II (Requirements & Project Management)

- Functional vs Non-functional requirements
- Software Requirements Specification (SRS)
- Requirements engineering process
- Requirements elicitation and analysis
- Requirements validation
- Requirements management
- Risk management
- Reactive vs Proactive risk strategies
- Risk identification and projection
- RMMM plan
- Software project planning
- Estimation techniques
- Project scheduling
- Agile planning

### UNIT – III (Design, Testing & Metrics)

- Design process and design quality
- Design concepts
- Software architecture
- Data design
- UML diagrams (class, sequence, use case)
- Component diagrams
- Testing strategies
- Black-box testing
- White-box testing

- Validation testing
- System testing
- Debugging
- Software metrics

---

## UNIT – IV (Quality & Maintenance)

- Software quality concepts
- Software quality assurance (SQA)
- Software reviews
- Formal technical reviews
- Statistical quality assurance
- Software reliability
- Release management
- Release planning and strategies
- Risk management in release
- Software maintenance
- Migration strategies