JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

B.Tech.-II SEMESTER | T3 EXAM - MAY 2026

COURSE CODE (CREDITS): 25B11CI211 (4) MAX. MARKS: 35

COURSE NAME: Software Development Fundamentals - II (SDF-II). MAX. TIME: 2 Hours

COURSE INSTRUCTORS: A. Kumar, D. Gupta (Coord.), F. Firdous, H. Srivastava, N. Singla, and P. Aar

***Note:** 1) All questions are compulsory. Marks and COs for each question are indicated. 2) **Answer the questions in the given order.** 3) **Be concise and write neatly.** 4) Use of calculators is not allowed.*

Q. No.	Question	CO	Marks
Q. 1	A student's academic information is maintained using a class StudentProfile, which includes attributes such as CGPA, number of completed projects, and internship performance. For quick evaluation during campus recruitment, this detailed data is converted into a simplified employability score or rating (e.g., Excellent, Good, Average, Poor), represented by another class PlacementReport. Design both classes in C++ and implement conversion routines (such as conversion functions or constructors) to convert a StudentProfile object into a PlacementReport object.	1	5
Q. 2	Design a university system that models student success using a UML-based composition model. Create a base class named Achievement with a string title, and derive two specialized classes: AcademicHonors (containing double cgpa) and LeadershipRole (containing int teamSize). Each derived class should include a member function getValue() that returns a calculated score based on its specific data. Using composition, design a StudentLeader class that 'has' objects of both AcademicHonors and LeadershipRole as private data members. Additionally, implement a member function showTotalImpact() within the StudentLeader class that aggregates the scores to compute and display a single total impact score.	5	5
Q. 3	Construct a generic class template named Pair<T> designed to store two private data members of a placeholder type T. Implement the following public member functions: • getLarger() compares the two stored values and returns the maximum. • swapValues() swaps the contents of the two members. Further, identify and explain a scenario where a syntactically perfect class template fails to compile during instantiation.	2	5
Q. 4	Create a C++ class template Stack using an array of size 5 to represent the stack. The class should support the following operations: push() to insert an element into the stack, pop() to delete an element from the stack, and display() to show all the elements present in the stack. Also, write appropriate logic to handle stack overflow	3	5

	and underflow conditions during insertion and deletion operations. In the main() function, invoke the member functions on a list of integers and strings.		
Q. 5	Implement a singly linked list using a class named LinkedList in C++ to store integer data values. The implementation should support the following basic operations: inserting a new node at the beginning of the list, inserting a new node at the end of the list, deleting the first node of the list, and traversing the list to display all its elements in sequence.	3	5
Q. 6	a) Explain how the use of function overloading and templates can lead to code bloating in a C++ program. b) Analyze the role of a tail pointer in a singly linked list. Evaluate its impact on the time complexity of insertion and deletion operations at the end of the list. c) Briefly explain iterators, the auto keyword, and range-based loops in C++ in terms of their usage and readability. d) Recursion consumes more memory compared to iteration. Explain it briefly with reference to call stack and stack unwinding during recursive function execution.	2,	[1+2 +2+1 =6]
Q. 7	For each program below (a-c), state the output and **briefly explain** your answer in 1-2 sentences. Assume the following statements are already included: #include <iostream> using namespace std;	4	[1*4 = 4]

a)

```
template <class T> class CTest {
public:
   static int iCount;
   CTest() {  cout << iCount++;  }
};
template <class T> int CTest <T> :: iCount = 1;
int main() {
   CTest <int> obj1;
   CTest <float> obj2;
   CTest <int> obj3;
   CTest <float> obj4;
   return 0; }
```

b)

```
int main() {
   int queue[3], front = -1, rear = -1;
   front = rear = 0; queue[rear] = 5;
   if (front == -1) cout << "Underflow ";
   else {
      cout << queue[front] << " ";
      front = rear = -1; }
   if (front == -1) cout << "Empty";
   return 0;}
```

c)

```
void displayRec(int count) {
   if (count == 0) return;
   cout << count * 2;
   displayRec(count - 1);
   cout << count + 5; }
int main() {
   displayRec(1);
   return 0;}
```

d) Determine time complexity using Big O notation of the following code snippet with proper justification:

```
void analyze(int n) {
   for (int i = 0; i < n*n; i++) {
      cout << i; }
   for (int i = 0; i < n*n; i++) {
      for (int j = 0; j < n*n*n; j++) {
         cout << i + j; }
   }}
```