



International Islamic University Chittagong
Department of Computer Science & Engineering
B.Sc. in CSE, Mid Term Examination, Spring 2025
Course Code: CSE-1221 Title: Computer Programming 2
Total Marks: 30 Time: 90 minutes

Answer all 3 from the following Questions. Figures in the right-hand margin indicate full marks

Answers must be done sequentially

- | # | | Marks | COs | DL |
|-----|--|-------|-----|----|
| 1 | a) Describe the characteristics of OOP with examples. | 3 | CO1 | C1 |
| | b) Write a class called 'Book' having two private member variables: 'title' and 'author'.
- Define a parameterized constructor to initialize them.
- In the 'main()' function, create two objects and display their data using a member function 'display_info'. | 2 | CO1 | C3 |
| | c) Identify and explain the error in the following program: | 2 | CO1 | C2 |
| | <pre>class Student { int a = 10; float g; public: void display() { cout << a << endl; } }; main() { Student st; st.g = 3.75; st.display(); }</pre> | | | |
| | Or, | | | |
| | <pre>class Student { int id = 101; public: float grade; void display(); }; void display() { cout << id << endl; } main() { Student st; st.grade = 4.0; st.display(); }</pre> | | | |
| d) | Explain the difference between a structure and a class in C++. Why is class preferred for real-world programming? Use appropriate code or examples. | 3 | CO1 | C2 |
| or, | Describe what problems associated with structured programming are solved by object-oriented programming and how? | | | |
| 2 | a) What is automatic inline function? When should we use inline and when shouldn't? Give an example. | 2 | CO1 | C2 |
| or, | Compare and explain access specifiers: public, private, and protected. | | | |
| b) | Correct the following code and explain the reason: | 3 | CO1 | C2 |
| | <pre>#include<iostream> using namespace std; class Demo { int num; public: Demo(int num) { this.num = num; cout << "Constructor: " << num << endl; } ~Demo() { cout << "Destructor: " << num << endl; } }; int main() { Demo d1(10); }</pre> | | | |

```
Demo d2(20);
{ Demo d3(30); }
cout << "Inside main function\n";
return 0; }
```

- c) Write a class named **Item** with private members: **item_id**, **item_name**, **manufacturer**, **and price**. Include both default and parameterized constructors, and a **set_item()** function using the this pointer to assign values from user input. Create an array of objects in **main()**, take input for n items, and display details of a specific item by matching **item_id**. Also, write a non-member function named **ApplyDiscount()** to modify the price of an Item and demonstrate it in **main()**. 5 CO2 C3
- 3 a) Why would someone overload a function? How can a compiler identify an overloaded function? Describe using an example. 3 CO2 C2
- or, Write a program to demonstrate **constructor overloading** and explain how different constructors are invoked for different objects.
- b) #include <iostream> 5 CO2 C3
using namespace std;
void display(int a = 5, int b = 10) {
 cout << "a = " << a << ", b = " << b << endl;
}
void display (double x) {
 cout << "x = " << x << endl;
}
int main() {
 display();
 display(15);
 display(15, 25);
 display(10.5);
 return 0;
}
- i) Identify and explain the **output** of the above program.
- ii) What kind of **overloading ambiguity** could arise if we add another **display(int)** function? Justify your answer.
- iii) Rewrite the above program **3b(ii)** by removing ambiguity, ensuring function overloading works correctly with **default arguments**.
- c) Why we use copy constructor? Write an example of copy constructor. 2 CO1 C1