


2.5 * 7.5 = 10.5 / 13

	National University of Computer and Emerging Sciences (Lahore)		
	Course:	OOP	Course code: CS217
	Section:	BSCS-2E	Semester: Spring 2024
	Duration:	40 minutes	Total Marks: 10
	Date:	7-5-24	ID: [REDACTED] 4B
	Name:	[REDACTED]	Roll no: [REDACTED]

Question 1:

NOTE: Read the entire question first before attempting.

A software company wants to develop a system to manage different types of employees. Implement a system that categorizes employees into three classes: Manager, Developer, and Quality Analyst. Ensure that all employees must belong to one of the specialized types, and no generic employee is created in the system at any point. Each type of employee, derived from a base class, has some shared and some unique attributes and/or behaviors as described below:

1: Manager:

Base

- Attributes: name (string), age (int), salary (double), department (string)
- Behaviors: void conductMeeting() → prints "Conducting a meeting in {department}".

2: Developer:

- Attributes: name (string), age (int), salary (double), programmingLanguage (string)
- Behaviors: void writeCode() → prints "Writing code in {programmingLanguage}".

3: Quality Analyst:

- Attributes: name (string), age (int), salary (double), testingTool (string)
- Behaviors: void testCode() → prints "Testing code using {testingTool}".

1. Implement default and parameterised constructors, destructors and a polymorphic function performtask() method in each derived class to define the task of each employee as described in behaviours.
2. Give output of the main given on the next page.

```

#include <iostream>
using namespace std;

// Implement the classes here (on the sheet provided)

int main() {
    // Create employee objects
    Employee* manager = new Manager("John Doe", 40, 80000.0, "IT");
    Employee* developer = new Developer("Jane Smith", 35, 60000.0, "C++");
    Employee* qualityAnalyst = new QualityAnalyst("Mike Johnson", 45, 55000.0,
"Selenium");

    // Perform duties
    manager->performTask();
    developer->performTask();
    qualityAnalyst->performTask();

    // Free memory
    delete manager;
    delete developer;
    delete qualityAnalyst;

    return 0;
}

```

Output:

Conducting a meeting in IT ✓
 Writing code in C++ ✓
 Testing code using Selenium ✓

③