|  |  |  |
| --- | --- | --- |
| **Object-oriented Programming (CS-1004)** | **Sessional-II Exam** | |
| Date: 7/4/2025 | **Total Time:** | 1hr |
| **Course Instructor(s)** | **Total Marks:** | **30** |
| Basit Ali, Minhal, Sumaiya, Nida, Sobia, Abeeha, Atiya, Abeer, Bakhtawar. | **Total Questions**: | 02 |
| Roll No Section Student Signature | |  |

**Do not write below this**

# Attempt all questions.

***CLO 1:*** *Discuss knowledge of underlying concepts of object-oriented paradigm like abstraction, encapsulation, polymorphism, inheritance etc.* 

**[Marks 10, 15 min]**

Q1 : Answer the following questions.

1. Can a friend function be overridden in a derived class, and if not, why?
2. Why does 5 + obj fail for a user - defined class, while obj + 5 works when overloading operator + as a member function?
3. If multiple objects of a class are created, why do they all share the same static member but have separate non - static members?
4. **Parent \*obj = new Child ( ); obj -> performTask( );**

Can you tell by looking at only these lines is this runtime polymorphism or static polymorphism?

1. Consider a class XYZ that has both a member function XYZ operator + (XYZ obj) and a global function XYZ operator + (XYZ obj1, XYZ obj2) Will the below code create an ambiguity?

XYZ obj1, obj2;

XYZ obj3 = obj1 + obj2;

***CLO 4:*** *Illustrate Object-Oriented design artifacts and their mapping to Object-Oriented Programming using C++.*

**[Marks 20, 45 min]**

Q2: You are tasked with designing an AI-powered smart car system that consists of multiple AI modules working together to enhance driving capabilities.

**Note: Program each part separately, do not merge them together in classes- you can comment which class the code snippets belong to when attempting each part.**

1. The system includes a base class AIComponent, which serves as the core AI framework. The car has different AI subsystems; NavigationAI which handles GPS and route planning has data members start, destination and estimated\_time, EntertainmentAI which manages music and voice interactions has data members volume and song playlist and SelfDrivingAI which combines navigation and entertainment while making autonomous driving decisions has a data member current\_speed. Each AI module should have a function performTask() to demonstrate dynamic polymorphism. There need not be any function logic, but it should print the task that the AI is performing. (3 Marks)
2. NavigationAI has a function routeMap that accepts a starting point(string) and destination(string) to plan a route towards the destination (It must print a message “Starting navigation from Source to Destination. Drive safely”). SelfdrivingAI should override this function to also ask the user their entertainment choices, and then invoke the routeMap method present in NavigationAI. (2 Marks)
3. Create a function demonstratePolymorphism() : Make an array of AIComponent\* pointers (in this function) to store different AI modules(one of each type of AI) and call performTask() dynamically. (4 marks)
4. compareRoutes(NavigationAI, NavigationAI) – A friend function that takes two NavigationAI objects as arguments and checks if they have identical routes. If both objects have the same start and destination, it returns true, otherwise false. (2 marks)
5. RemoteAIControl (Friend of SelfDrivingAI) – A specialized class that acts as an external override mechanism for SelfDrivingAI. It allows a remote operator to modify the speed of the car to ensure passenger safety, or to update the destination of the car. (3 marks)
6. You are required to implement operator overloading that enables the following operations to work in the main function. (3 x 2 = 6 marks)

int main() {

NavigationAI n1;

SelfDrivingAI s1;

EntertainmentAI e1,e2;

n1\*5; //updates estimated\_time of n1 to estimated\_time\*5;

if(s1==n1) cout << “Routes are identical”;

e1+=e2; //Merges e2’s playlist into a e1

}