**COLLEGE OF BUSINESS EDUCATION**

**DAR ES SALAAM CAMPUS**

**COURSE:** BACHELOR OF INFORMATION TECHNOLOGY

**MODULE NAME:** OBJECT ORIENTED PROGRAMMING IN C++

**MODULE CODE:** ITU07209

**LECTURE NAME:** AHMED KIJAZI

**STUDENT'S NAME: MBARUKU HEMED**

**REGISTRATION NUMBER:** 03.2481.01.01.2023

**TASK:** INDIVIDUAL ASSIGNMENT

**SUBMISSION DATE: 22/04/2024**

**Question 1:**

Brief explain the following terms as they are being used in C++ programming

a) Object

b) Class

c) Encapsulation

d) Polymorphism

**Question 2:**

Write a single C++ program that implement the concept of object, functions and class

**QN 1(a):**

**Objects:** An object is an instance of a class. It represents a specific entity or instance based on the class definition. When an object is created, memory is allocated to store its data members, and it can invoke the methods defined by its class. Objects have their own unique state, but they share the common structure and behavior defined by their class. For example, if a class represents a "Car," an object of that class can represent a specific car instance with its own set of properties and behaviors.

**(b):**

**Classes:** A class is a blueprint or template that defines the structure and behavior of objects. It serves as a blueprint for creating objects of a specific type. A class encapsulates data (in the form of data members) and behaviors (in the form of member functions or methods) that are common to all objects of that class. It defines the properties (attributes) and actions (methods) that objects of the class can possess. In simpler terms, a class defines the characteristics and behaviors that objects of that class will exhibit.

**(c):**

**Encapsulation:** Encapsulation is the process of bundling data and the methods (or functions) that operate on that data within a single unit called an object. It involves hiding the internal details and providing a well-defined interface for interacting with the object. The main idea behind encapsulation is to ensure data integrity and control access to the object's properties by enforcing the principle of information hiding. By encapsulating data and methods together, objects can protect their internal state and expose only the necessary operations to the outside world, promoting modular and secure code.

**(d):**

**Polymorphism:** Polymorphism refers to the ability of objects of different classes to respond to the same message or method invocation in different ways. It allows objects to exhibit different behaviors based on their specific class or type. Polymorphism can be achieved through method overriding and method overloading. Method overriding occurs when a subclass provides its own implementation of a method defined in its superclass. This allows the subclass to customize the behavior of the inherited method. Method overloading, on the other hand, involves defining multiple methods with the same name but different parameters within a single class.

**QN2:**

**#include<iostream>**

**using namespace std;**

//create class called TriAngle

**class TriAngle**

**{**

**private:**

//member decleare as private are accessible only from the within the same class where they are decleare

//private members can not be accessed or manipulate directly by code outside the class

**double base;**

**double heigth;**

**public:**

//member decleare as public are accessable from outside the class include other class and function.

//public member can be accessed and manipulate directly by code that has access to the to the objective of the class

**TriAngle(double B, double H)**

**{**

**base=B;**

**heigth=H;**

**}**

//function to set base of the triangle

**void setbase(double B)**

**{**

**base = B;**

**}**

//function to set heigth of the triangle

**void setheigth(double H)**

**{**

**heigth=H;**

**}**

//function to calculate area of triangle

**double calculateArea()**

**{**

**return 0.5\*base\*heigth;**

**}**

**};**

**int main()**

**{**

**double base,heigth;**

//prompt the user to calculate base and heigth of the area of triangle

**cout<<"enter the base: "<<endl;**

**cin>>base;**

**cout<<"enter the heigth: "<<endl;**

**cin>>heigth;**

//create an objective of the triangle class ith the user input

**TriAngle triangle(base,heigth);**

//print the area of the triangle

**cout<<"the area of triangle are: "<<triangle.calculateArea()<<endl;**

**return 0;**

**}**

**Explanation**:

* I define CLASS called ‘**TriAngle**’, which has private data members **‘base’** and **‘heigth’**.
* The class has a constructor to I nitialize the **‘base’** and **‘heigth’** of the triangle.
* It also has public member functions **‘setbase()’** and **‘setheigth()’** to set the base and height of the triangle respectively.
* Additionally it has a public member function **‘calculateArea()’** to calculate the the area of triangle.
* In the **‘main()’** function, we prompt the user to enter the base and heigth of the triangle.
* Then I create the OBJECT **‘triangle’** of the ‘**TriAngle**’ class with the user input values.
* Finally I call the ‘**calculateArea()**’ function on the **‘triangle’** object to calculate the area of the rectangle and print the result.