Bangladesh University
Mid Term Exam

Student Name: Md. Ramjan Hosen
Student ID: 202011056161
Department: Computer Science and Engineering
Batch: 58th
Subject: CSE-1305 [Object Oriented Programming]
Submitted to Khan Md. Hasib

Answer to the question no: 01

What is class in C++?
Ans: A class in C++ is the building block, that leads to Object-Oriented programming. It is a user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class. A C++ class is like a blueprint for an object.

Example:
C++ is an object-oriented programming language.
Everything in C++ is associated with classes and objects, along with its attributes and methods. For example: in real life, a car is an object. The car has attributes, such as weight and color, and methods, such as drive and brake.
Attributes and methods are basically variables and functions that belongs to the class. These are often referred to as "class members".
A class is a user-defined data type that we can use in our program, and it works as an object constructor, or a "blueprint" for creating objects.

To create a class, use the class keyword:
Create a class called "MyClass":

```
class MyClass {       // The class
  public:             // Access specifier
    int myNum;        // Attribute (int variable)
    string myString;  // Attribute (string variable)
};
```

Example explained:

- The class keyword is used to create a class called MyClass.
- The public keyword is an access specifier, which specifies that members (attributes and methods) of the class are accessible from outside the class.

- Inside the class, there is an integer variable myNum and a string variable myString. When variables are declared within a class, they are called attributes.
- At last, end the class definition with a semicolon ;.

What is object in C++?

Ans: This is the basic unit of object oriented programming. An object is an entity that has state, behavior and property. Objects are used to model real world entities that we find in the daily life. That is both data and function that operate on data are bundled as a unit called as object. Every object will have data structures called attributes and behavior called operations.

Example:

In C++, an object is created from a class. We have already created the class named MyClass, so now we can use this to create objects.

To create an object of MyClass, specify the class name, followed by the object name.

To access the class attributes (myNum and myString), use the dot syntax (.) on the object:

Create an object called "myObj" and access the attributes:

```
class MyClass {       // The class
  public:             // Access specifier
    int myNum;        // Attribute (int variable)
    string myString;  // Attribute (string variable)
};

int main() {
  MyClass myObj;  // Create an object of MyClass

  // Access attributes and set values
  myObj.myNum = 20;
  myObj.myString = "Some text";

  // Print attribute values
  cout << myObj.myNum << "\n";
  cout << myObj.myString;
  return 0;
}
```

What is abstraction in C++?

Abstraction means displaying only essential information and hiding the details. Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation.

Example:

```
#include <iostream>
using namespace std;

class implementAbstraction
{
    private:
        int m, n;

    public:

        // method to set values of
        // private members
        void set(int x, int y)
        {
            m = x;
            n = y;
        }

        void display()
        {
            cout<<"m = " <<m << endl;
            cout<<"n = " << n << endl;
        }
};

int main()
{
    implementAbstraction obj;
    obj.set(15, 20);
    obj.display();
    return 0;
}
```

Output:

```
a = 15
b = 20
```

You can see in the above program we are not allowed to access the variables a and b directly, however one can call the function set() to set the values in a and b and the function display() to display the values of a and b.

Answer to the question no: 02

As we know both C and C++ are programming languages and used for application development. The main difference between both these languages is C is a procedural programming language and does not support classes and objects, while C++ is a combination of both procedural and object-oriented programming languages.

The following are the important differences between C and C++.

C	C++
1. C was developed by Dennis Ritchie between the year 1969 and 1973 at AT&T Bell Labs.	1. C++ was developed by Bjarne Stroustrup in 1979.
2. C does no support polymorphism, encapsulation, and inheritance which means that C does not support object oriented programming.	2. C++ supports polymorphism, encapsulation, and inheritance because it is an object oriented programming language.
3. C is a subset of C++.	3. C++ is a superset of C.
4. C contains 32 keywords.	4. C++ contains 63 keywords.
5. For the development of code, C supports procedural programming.	5. C++ is known as hybrid language because C++ supports both procedural and object oriented programming paradigms.
6. Data and functions are separated in C because it is a procedural programming language.	6. Data and functions are encapsulated together in form of an object in C++.
7. C does not support information hiding.	7. Data is hidden by the Encapsulation to ensure that data structures and operators are used as intended.
8. Built-in data types is supported in C.	8. Built-in & user-defined data types is supported in C++.
9.C is a function driven language because C is a procedural programming language.	9. C++ is an object driven language because it is an object oriented programming.
10. Function and operator overloading is not supported in C.	10. Function and operator overloading is supported by C++