

Department of Computer Science

Mid Term Summer 2020

Subject: Object Oriented Programming

Time: 180 mins

BS (CS,SE)

Instructor: M.Ayub Khan

There are total **3** questions in this paper.

Max Marks: 30

Note:

At the top of the answer sheet there must be the ID, Name and semester of the concerned Student.

Students must have to provide the output of their respective programs. Students have same answers or programs will be considered fail. Programs in Python or codes should be explained clearly.

As this paper is online so incase of any ambiguity my Whatsapp no. is 034499121116.

**Each question carry equal marks.
Please answer briefly.**

Name : shayan khan
Id : 6833
Subject : Object Oriented Programming
Submitted To : Sir M.Ayub Khan

Q1. What is Class and role of object in a Class, explain in detail with the help of a suitable program ?

Answer:

Class:

In object-oriented programming, a class is an extensible program-code-template for creating objects, providing initial values for state (member variables) and implementations of behavior (member functions or methods). In many languages, the class name is used as the name for the class (the template itself), the name for the default constructor of the class (a subroutine that creates objects), and as the type of objects generated by instantiating the class; these distinct concepts are easily conflated.

Create a Class

To create a class, use the class keyword:

Example

Create a class called " MyClass":

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

Role of object in a Class:

In object-oriented programming, a **class** is a blueprint for creating **objects** (a particular data structure), providing initial values for state (member variables or attributes), and implementations of behavior (member functions or methods).

The user-defined objects are created using the class keyword. The class is a blueprint that defines a nature of a future object. An **instance** is a specific object created from a particular class. Classes are used to create and manage new objects and support **inheritance** a key ingredient in object-oriented programming and a mechanism of reusing code.

Example

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 = 15;
    myObj.myString = "Some text";

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

Q2. Write a program about table printing which takes input from the user on the basis of OOP and explain in detail.

Answer:

```
num = int (input("Enter the num:"))

print ( " Multiplication table of num:  ")
for i in range (1, 11):
    print(num,"x",i, "=" num * i)
```

Output:

```
Enter the num :
Multiplication table of 6
 6 X 1 = 6
 6 X 2 = 12
 6 X 3 = 18
 6 X 4 = 24
 6 X 5 = 30
 6 X 6 = 36
 6 X 7 = 42
 6 X 8 = 48
 6 X 9 = 54
 6 X 10 = 6
```

Q3. Write a program about any 2 cars which can calculate the performance of both of them and explain in detail.

Answer:

```
file = open("cars.csv","r")
limit = int(input("Please enter the speed limit: "))
for line in file:
    details = line.split(",")
    speed = float(details[1])
    if speed > limit:
        print(details[0] + " " + details[1])
```

The finished program looks like this:

