Department of Computer Science

Semester Assignment Spring 2020

Subject: Object Oriented Programming

Time: 48hrs (9:00 am Monday (20th April) to 9:00 am Wednesday (22nd April))

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.

Student name :haider saleem id:16368 bs(cs)2

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

- **Object** − Objects have states and behaviors. Example: A dog has states - color, name, breed as well as behaviors – wagging the tail, barking, eating. An object is an instance of a class.
- **Class** − A class can be defined as a template/blueprint that describes the behavior/state that the object of its type support.

Classes in Java

A class is a blueprint from which individual objects are created.

Following is a sample of a class.

Example

```
public class Dog {
   String breed;
   int age;
   String color;

   void barking() {
   }

   void hungry() {
   }

   void sleeping() {
   }
}
```

A class can contain any of the following variable types.

- **Local variables** − Variables defined inside methods, constructors or blocks are called local variables. The variable will be declared and initialized within the method and the variable will be destroyed when the method has completed.
- **Instance variables** − Instance variables are variables within a class but outside any method. These variables are initialized when the class is instantiated. Instance variables can be accessed from inside any method, constructor or blocks of that particular class.
- **Class variables** − Class variables are variables declared within a class, outside any method, with the static keyword

Objects in Java

Let us now look deep into what are objects. If we consider the real-world, we can find many objects around us, cars, dogs, humans, etc. All these objects have a state and a behavior.

If we consider a dog, then its state is - name, breed, color, and the behavior is - barking, wagging the tail, running.

If you compare the software object with a real-world object, they have very similar characteristics.

Software objects also have a state and a behavior. A software object's state is stored in fields and behavior is shown via methods.

So in software development, methods operate on the internal state of an object and the object-to-object communication is done via methods.

An object consists of :

1. **State** : It is represented by attributes of an object. It also reflects the properties of an object.
2. **Behavior** : It is represented by methods of an object. It also reflects the response of an object with other objects.
3. **Identity** : It gives a unique name to an object and enables one object to interact with other objects.
4. **Declaring Objects (Also called instantiating a class)**
5. When an object of a class is created, the class is said to be **instantiated**. All the instances share the attributes and the behavior of the class. But the values of those attributes, i.e. the state are unique for each object. A single class may have any number of instances.

Object class is present in **java.lang** package. Every class in Java is directly or indirectly derived from the **Object** class. If a Class does not extend any other class then it is direct child class of **Object** and if extends other class then it is an indirectly derived. Therefore the Object class methods are available to all Java classes. Hence Object class acts as a root of inheritance hierarchy in any Java Program.

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

```java
import java.util.Scanner;
class MultiplicationTable
{
 public static void main(String args[])
 {
  int n, c;
  System.out.println("Enter an integer to print it's multiplication table");
  Scanner in = new Scanner(System.in);
  n = in.nextInt();
  System.out.println("Multiplication table of " + n);

  for (c = 1; c <= 10; c++)
   System.out.println(n + "*" + c + " = " + (n*c));
 }
}
```

output

```
E: \Java>Java MultipilcationTable
Enter an integer to print it’s multiplicationtable
9
Multipliction table of 9 is :-
```

```
9*1 = 9
9*2 = 18
9*3 = 27
9*4 = 36
9*5 = 45
9*6 = 54
9*7 = 63
9*8 = 72
9*9 = 81
9*10 = 90
E:\java_
```

First off all we import the scanner in the program from bultin library and we take input through the scanner then we creat a class of multiplication .we ceclar a variable to take input and store it in it .then we take input from the user and he we apply the loop which will start from 1 and check it it 10 and if the loop getout from 10 range the loop will sop and the result will be printed.

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

```java
//Algorithm for buying a car
//by comparing car price, MPG,
//local gas price, and the miles
//to be drive in total.
//idea from udacity Intro
//to Java course.


import java.util.Scanner;

public class buyingCars {

    public static void main (String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.println("Enter the price of car one: ");
        int carOnePrice = input.nextInt();

        System.out.println("Enter the price of car two: ");
        int carTwoPrice = input.nextInt();

        System.out.println("Enter the price of gas in your area: ");
        double gasPrice = input.nextDouble();

        System.out.println ("What is the average mpg of car one? ");
        int carOneMpg = input.nextInt();

        System.out.println ("What is the average mpg of car two? " );
        int carTwoMpg = input.nextInt();

        System.out.println("Give me an amount of miles total you think you will drive in this car: ");
        int milesDriven = input.nextInt();

        double gasCostCarOne = (milesDriven / carOneMpg) * gasPrice;
        double totalCostCarOne = gasCostCarOne + carOnePrice;

        double gasCostCarTwo = (milesDriven / carTwoMpg) * gasPrice;
        double totalCostCarTwo = gasCostCarTwo + carTwoPrice;

        if (totalCostCarOne < totalCostCarTwo) {

            System.out.println("Car One will, in total, cost less than car two. Buy it.");
        }

        else {

            System.out.println("Car two will, in total, cost less than car one. Buy it.");
        }
    }
}
```

For each car, compute. Gas cost = (miles driven / mpg) * price per gallon. Total cost = gas cost + car cost. If car 1 < car 2, buy car 1. Else, buy car 2.