PRACTICAL 1
CLASSES AND CONSTRUCTORS IN JAVA

Classes:

A class is a user defined blueprint or prototype from which objects are created. It represents the set of properties or methods that are common to all objects of one type. In general, class declarations can include these components, in order:

1. Modifiers: A class can be public or has default access (Refer this for details).
2. class keyword: class keyword is used to create a class.
3. Class name: The name should begin with an initial letter (capitalized by convention).
4. Superclass(if any): The name of the class's parent (superclass), if any, preceded by the keyword extends. A class can only extend (subclass) one parent.
5. Interfaces(if any): A comma-separated list of interfaces implemented by the class, if any, preceded by the keyword implements. A class can implement more than one interface.
6. Body: The class body surrounded by braces, { }.

The basic syntax is

```
public class <classname> {
   public static void main(String []args) {
      <code>
   }
}
```

Constructors:

Java constructors is a terminology been used to construct something in our programs. A constructor in Java is a special method that is used to initialize objects. The constructor is called when an object of a class is created. It can be used to set initial values for object attributes.

In Java, a constructor is a block of codes similar to the method. It is called when an instance of the class is created. At the time of calling the constructor, memory for the object is allocated in the memory. It is a special type of method which is used to initialize the object. Every time an object is created using the new() keyword, at least one constructor is called. Constructors are different from methods in these ways:

1. Constructors must have the same name as the class within which it is defined while it is not necessary for the method in Java.
2. Constructors do not return any type while method(s) have the return type or void if does not return any value.
3. Constructors are called only once at the time of Object creation while method(s) can be called any number of times.

The syntax is:

```
public class <classname> {
  public <classname>() {
    <code>
  }

  public static void main(String[] args) {
    <classname> <objname> = new <classname>();
    <code>
  }
}
```

Q1. Write a java program to Create a class named Calculate and find area and perimeter a rectangle

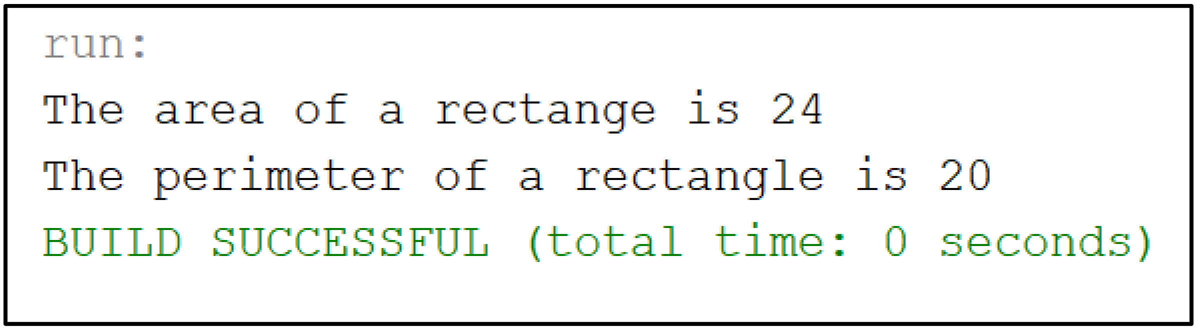
```
package practical1;

public class calculate {
    public void area()
    {
    int l,b,area;
    l = 4;
    b = 6;
    area = l * b;
    System.out.println("The area of a rectange is " + area);
    }

    public void peri()
    {
    int l,b,peri;
    l = 4;
    b = 6;
    peri = (2*l) + (2*b);
    System.out.println("The area of a perimeter is " + peri);
    }

    public static void main(String[] args) {
        calculate c1=new calculate();
        c1.area();
        c1.peri();
    }

}
```



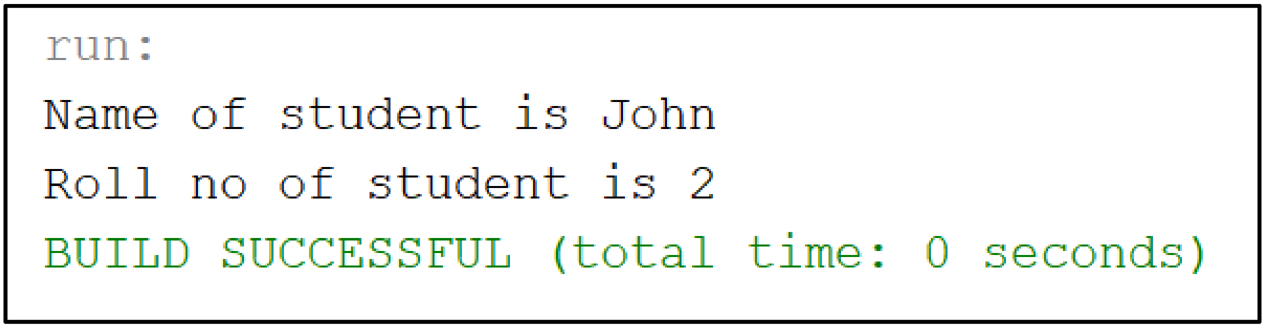
```
run:
The area of a rectange is 24
The perimeter of a rectangle is 20
BUILD SUCCESSFUL (total time: 0 seconds)
```

Q2. Create a class named "Student" with String variable "name" and integer variable "roll_no". Assign the value of roll_no as "2" and that of name as "John" by creating an object of the class Student.

```
package practical1;

public class Student {
    int roll_no;
    String name;

    public static void main(String[] args){
        Student s1=new Student();
        s1.roll_no = 2;
        s1.name = "John";
        System.out.println("Name of student is " + s1.name);
        System.out.println("Roll no of student is " + s1.roll_no);
    }
}
```



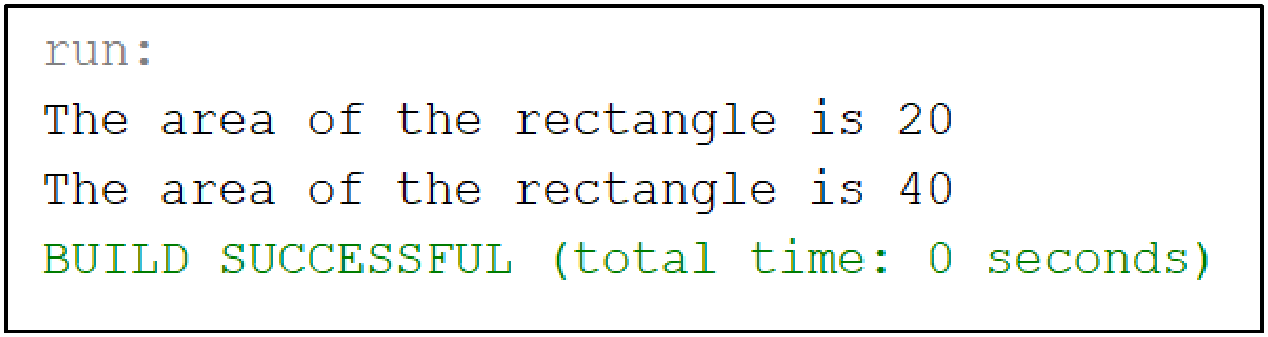
```
run:
Name of student is John
Roll no of student is 2
BUILD SUCCESSFUL (total time: 0 seconds)
```

Q3. Write a program to print the area of two rectangles having sides (4,5) and (5,8) respectively by creating a class named “Rectangle” with a method named “Area” which returns the area and length and breadth passed as parameters to its constructor.

```java
package practical1;

public class Rectangle{
    public void Area(int length, int breadth){
        int a;
        a = length * breadth;
        System.out.println("The area of the rectangle is " + a);
    }

    public static void main(String[] args){
        Rectangle a1=new Rectangle();
        a1.Area(4, 5);
        a1.Area(5, 8);
    }
}
```



```
run:
The area of the rectangle is 20
The area of the rectangle is 40
BUILD SUCCESSFUL (total time: 0 seconds)
```

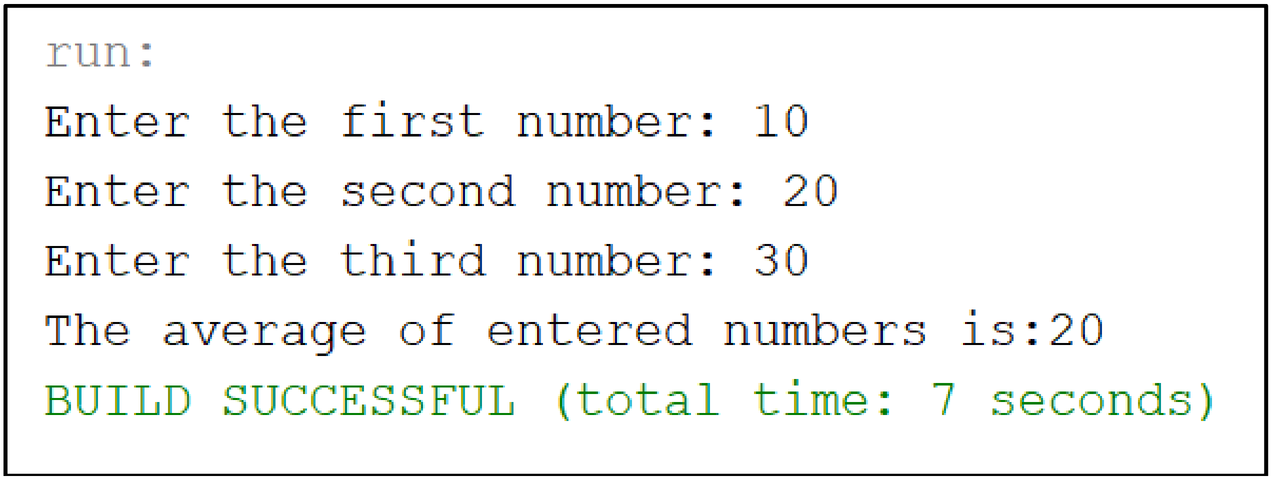
Q4. Print the average of three numbers entered by user by creating a class named “Average” having a method to calculate and print the average.

```
package practical1;
import java.util.Scanner;

public class Average {
    public void calculate(int a, int b, int c){
        int avg = (a + b + c ) / 3;
        System.out.println("The average of entered numbers is:" +
avg);
    }

    public static void main(String[] args){
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter the first number: ");
        int num1 = scan.nextInt();
        System.out.print("Enter the second number: ");
        int num2 = scan.nextInt();
        System.out.print("Enter the third number: ");
        int num3 = scan.nextInt();
        scan.close();

        Average a1=new Average();
        a1.calculate(num1,num2,num3);
    }
}
```



```
run:
Enter the first number: 10
Enter the second number: 20
Enter the third number: 30
The average of entered numbers is:20
BUILD SUCCESSFUL (total time: 7 seconds)
```

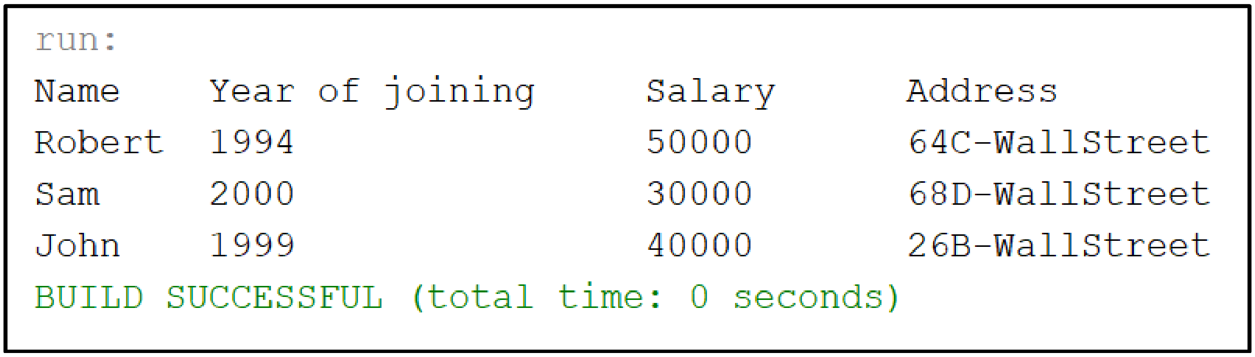
Q5. Write a program that would print the information (name, year of joining, salary, address) of three employees by creating a class named "Employee" The output should be as follows:

Name	**Year of joining**	**Address**
Robert	**1994**	**64C- WallsStreat**
Sam	**2000**	**68D- WallsStreat**
John	**1999**	**26B- WallsStreat**

```
package practical1;

public class Employee {
    public void showData(String name, int yearofjoining, int
salary, String add){
        System.out.println(name + "\t" + yearofjoining + "\t\t    "
+ salary + "\t" + add);
    }

    public static void main(String[] args){
        System.out.println("Name     Year of joining       Salary
Address");
        Employee e1=new Employee();
        e1.showData("Robert", 1994, 50000, "64C-WallStreet");
        e1.showData("Sam", 2000, 30000, "68D-WallStreet");
        e1.showData("John", 1999, 40000, "26B-WallStreet");
    }
}
```



```
run:
Name    Year of joining       Salary      Address
Robert  1994                  50000       64C-WallStreet
Sam     2000                  30000       68D-WallStreet
John    1999                  40000       26B-WallStreet
BUILD SUCCESSFUL (total time: 0 seconds)
```