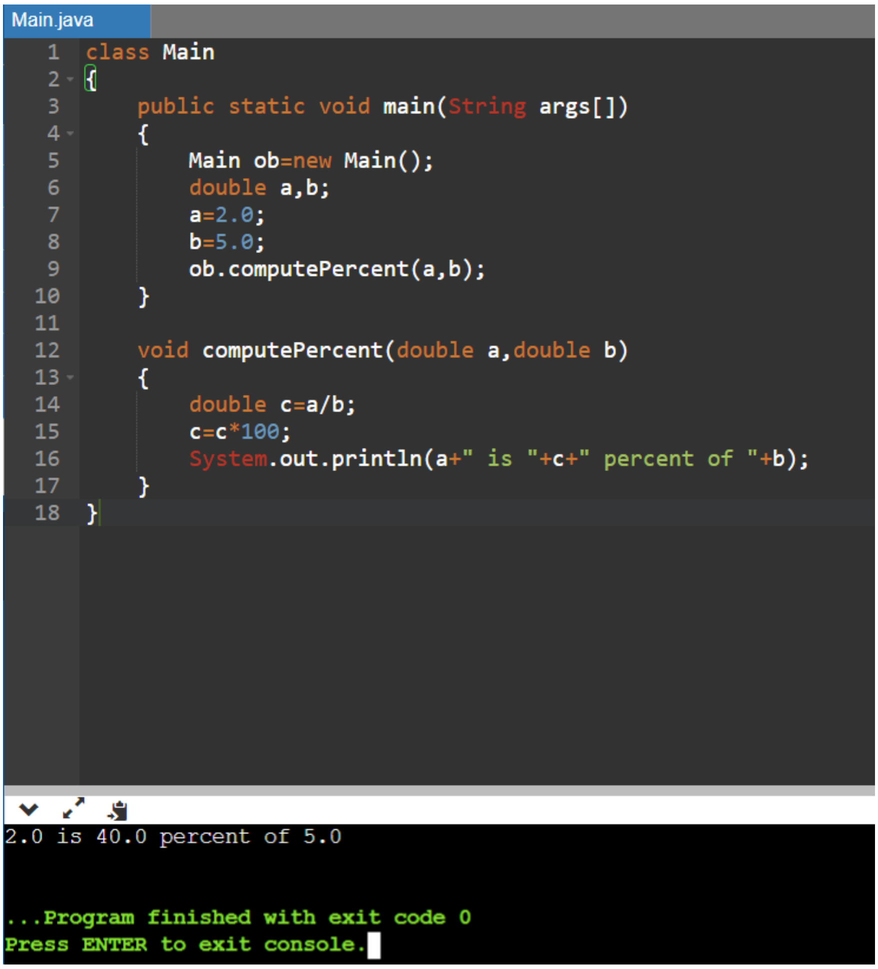
# LAB1

MAJJIGA JASWANTH 20BCD7171

1a question:

1.a. Create an application named Percentages whose main() method holds two double variables. Assign values to the variables. Pass both variables to a method named computePercent() that displays the two values and the value of the first number as a percentage of the second one. For example, if the numbers are 2.0 and 5.0, the method should display a statement similar to “2.0 is 40 percent of 5.0.” Then call the method a second time, passing the values in reverse order. Save the application as Percentages.java.

Code:

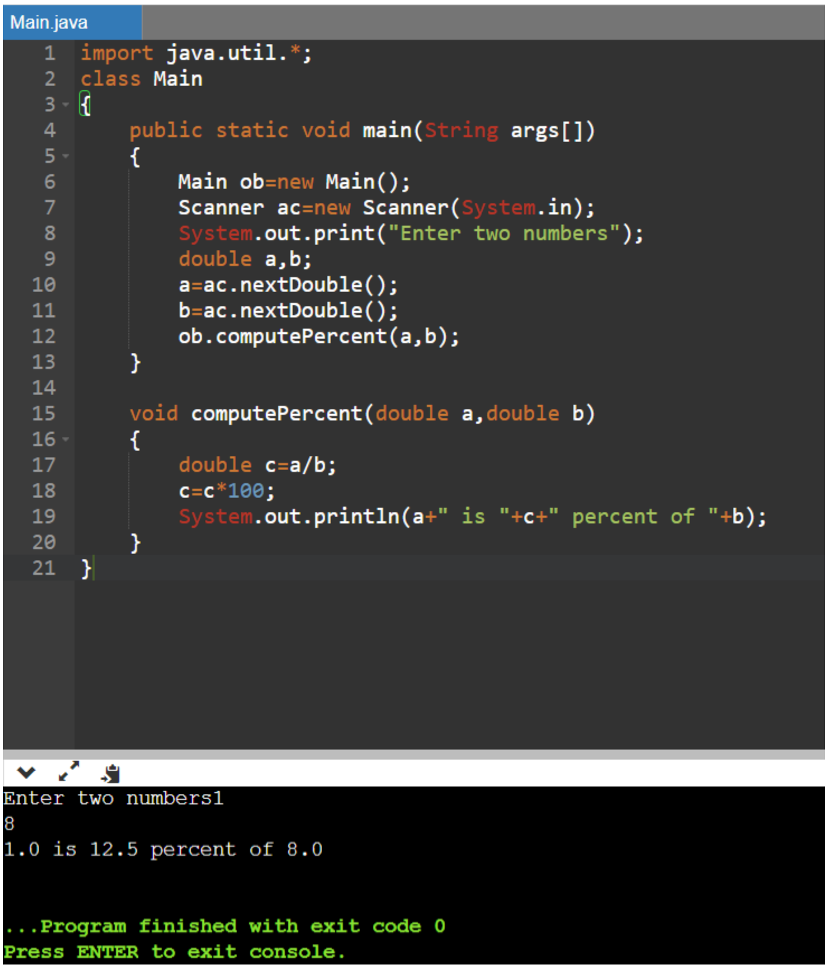


```java
class Main
{
    public static void main(String args[])
    {
        Main ob=new Main();
        double a,b;
        a=2.0;
        b=5.0;
        ob.computePercent(a,b);
    }

    void computePercent(double a,double b)
    {
        double c=a/b;
        c=c*100;
        System.out.println(a+" is "+c+" percent of "+b);
    }
}
```

```
2.0 is 40.0 percent of 5.0


...Program finished with exit code 0
Press ENTER to exit console.
```

1b question

Modify the Percentages class to accept the values of the two doubles from a user at the keyboard. Save the file as Percentages2.java.

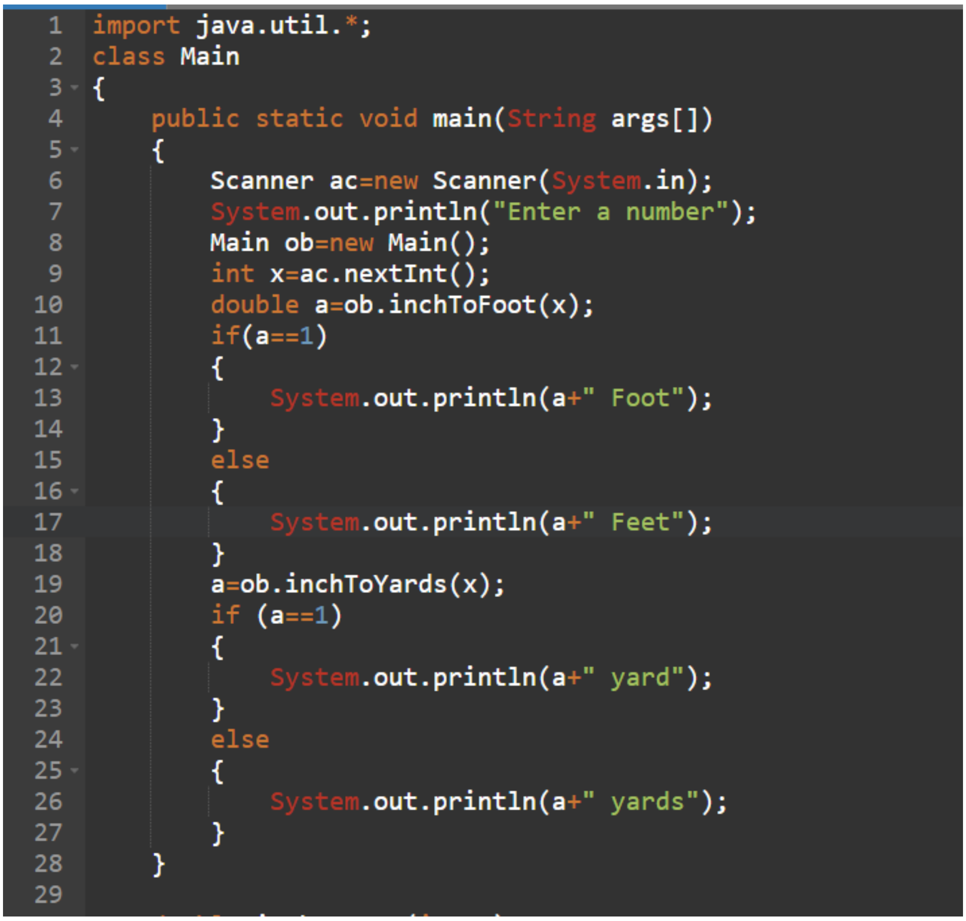


```java
import java.util.*;
class Main
{
    public static void main(String args[])
    {
        Main ob=new Main();
        Scanner ac=new Scanner(System.in);
        System.out.print("Enter two numbers");
        double a,b;
        a=ac.nextDouble();
        b=ac.nextDouble();
        ob.computePercent(a,b);
    }

    void computePercent(double a,double b)
    {
        double c=a/b;
        c=c*100;
        System.out.println(a+" is "+c+" percent of "+b);
    }
}
```

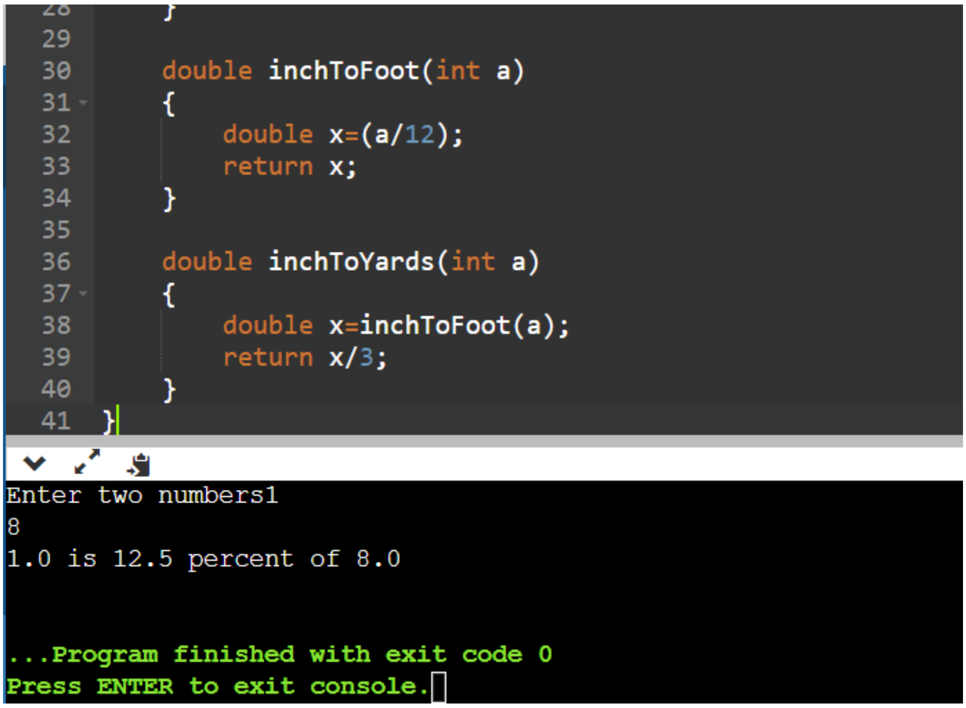
```
Enter two numbers1
8
1.0 is 12.5 percent of 8.0


...Program finished with exit code 0
Press ENTER to exit console.
```

Question 2:

**There are 12 inches in a foot and 3 feet in a yard. Create a class named InchConversion. Its main() method accepts a value in inches from a user at the keyboard, and in turn passes the entered value to two methods. One converts the value from inches to feet, and the other converts the same value from inches to yards. Each method displays the results with appropriate explanation. Save the application as InchConversion.java.**



```java
import java.util.*;
class Main
{
    public static void main(String args[])
    {
        Scanner ac=new Scanner(System.in);
        System.out.println("Enter a number");
        Main ob=new Main();
        int x=ac.nextInt();
        double a=ob.inchToFoot(x);
        if(a==1)
        {
            System.out.println(a+" Foot");
        }
        else
        {
            System.out.println(a+" Feet");
        }
        a=ob.inchToYards(x);
        if (a==1)
        {
            System.out.println(a+" yard");
        }
        else
        {
            System.out.println(a+" yards");
        }
    }

```



```java
    }

    double inchToFoot(int a)
    {
        double x=(a/12);
        return x;
    }

    double inchToYards(int a)
    {
        double x=inchToFoot(a);
        return x/3;
    }
}
```

```
Enter two numbers1
8
1.0 is 12.5 percent of 8.0


...Program finished with exit code 0
Press ENTER to exit console.
```

Question 3

**Assume that a gallon of paint covers about 350 square feet of wall space. Create an application with a main() method that prompts the user for the length, width, and height of a rectangular room. Pass these three values to a method that does the following:**

**•• Calculates the wall area for a room**

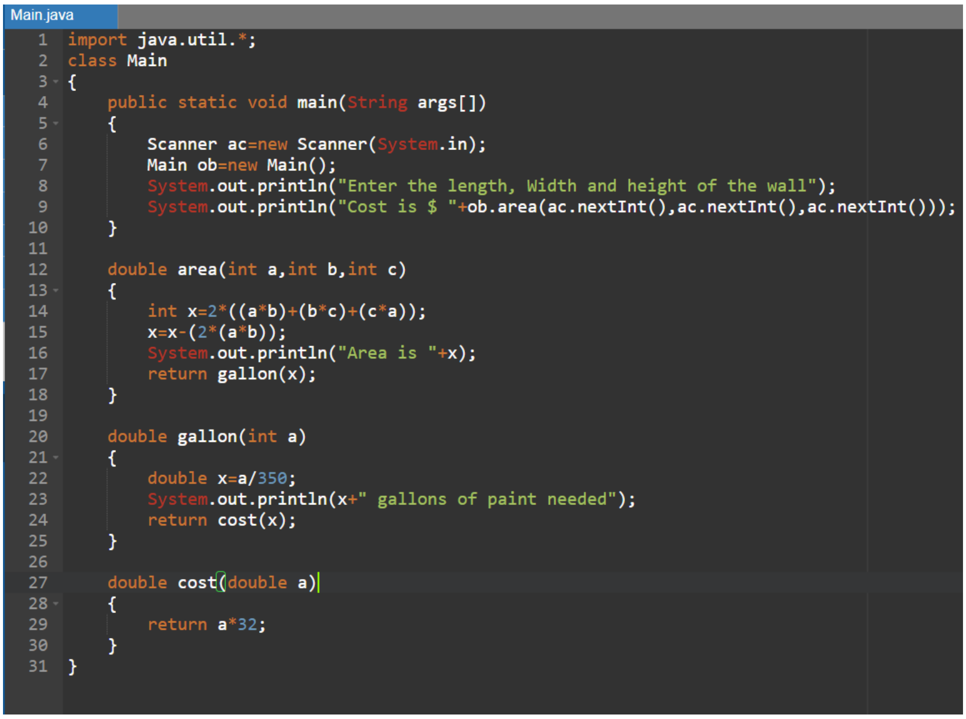
**•• Passes the calculated wall area to another method that calculates and returns the number of gallons of paint needed**

**•• Displays the number of gallons needed**

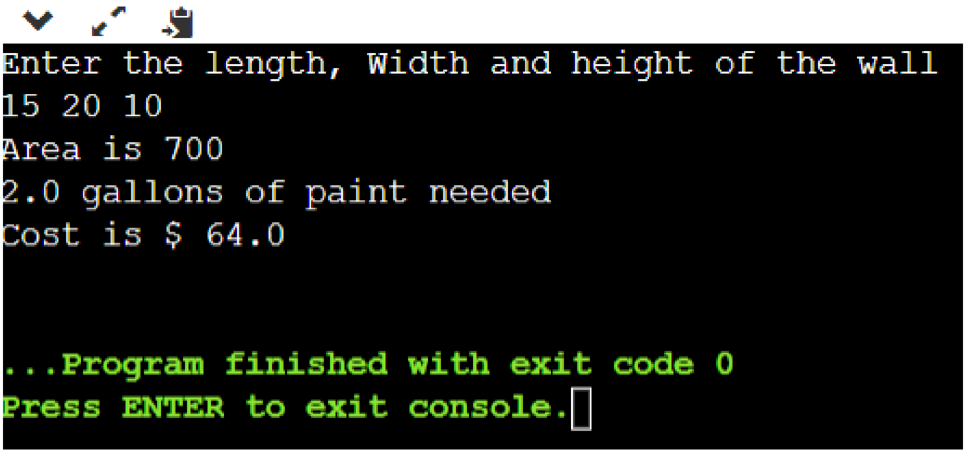
**•• Computes the price based on a paint price of $32 per gallon, assuming that the painter can buy any fraction of a gallon of paint at the same price as a whole gallon**

**•• Returns the price to the main() method**

**The main() method displays the final price. For example, the cost to paint a 15-by-20-foot room with 10-foot ceilings is $64. Save the application as PaintCalculator.java.**



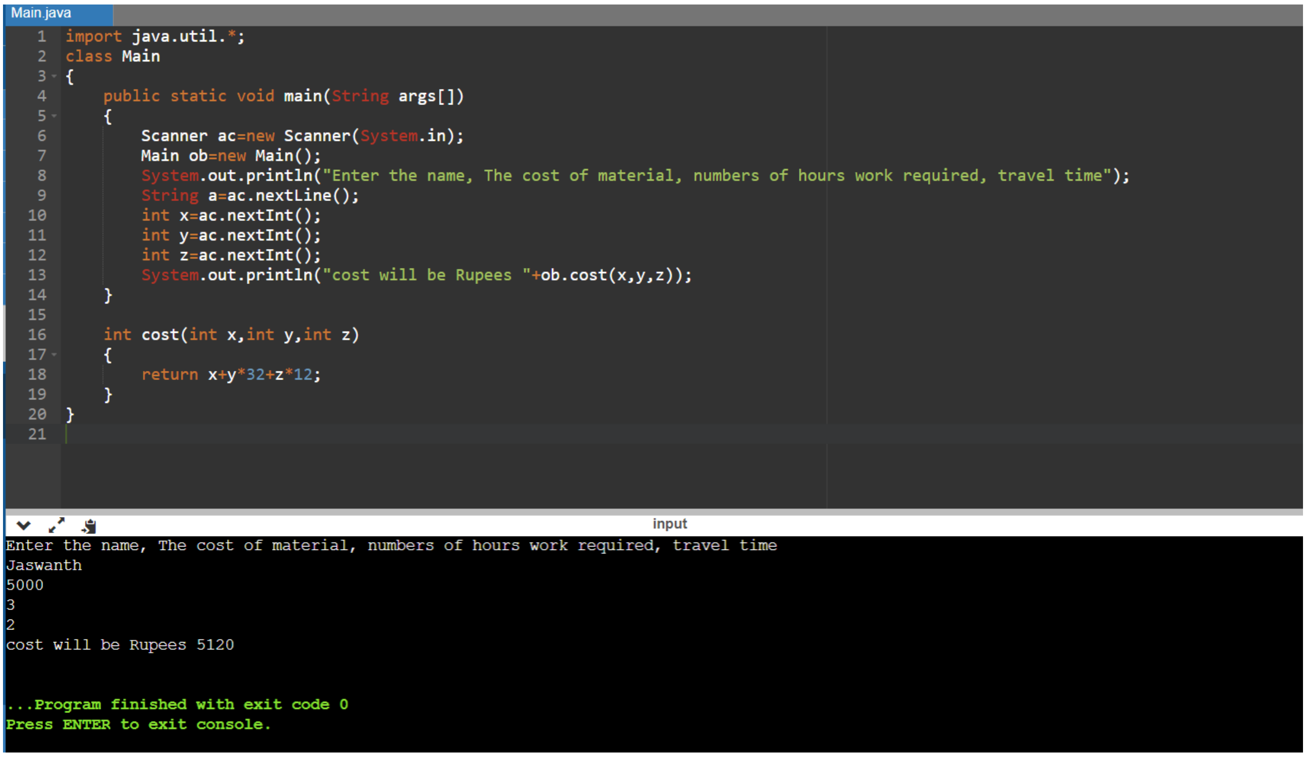
```java
import java.util.*;
class Main
{
    public static void main(String args[])
    {
        Scanner ac=new Scanner(System.in);
        Main ob=new Main();
        System.out.println("Enter the length, Width and height of the wall");
        System.out.println("Cost is $ "+ob.area(ac.nextInt(),ac.nextInt(),ac.nextInt()));
    }

    double area(int a,int b,int c)
    {
        int x=2*((a*b)+(b*c)+(c*a));
        x=x-(2*(a*b));
        System.out.println("Area is "+x);
        return gallon(x);
    }

    double gallon(int a)
    {
        double x=a/350;
        System.out.println(x+" gallons of paint needed");
        return cost(x);
    }

    double cost(double a)
    {
        return a*32;
    }
}
```



```
Enter the length, Width and height of the wall
15 20 10
Area is 700
2.0 gallons of paint needed
Cost is $ 64.0


...Program finished with exit code 0
Press ENTER to exit console.
```

Question 4

**Herbert's Home Repair estimates each job cost as the cost of materials plus $35 per hour while on the job, plus $12 per hour for travel time to the job site. Create a class that contains a main() method that prompts the user for the name of a job (for example, Smith bathroom remodel), the cost of materials, the number of hours of work required, and the number of hours travel time. Pass the numeric data to a method that computes estimate for the job and returns the computed value to the main() method where the job name and estimated price are displayed. Save the program as JobPricing.java.**

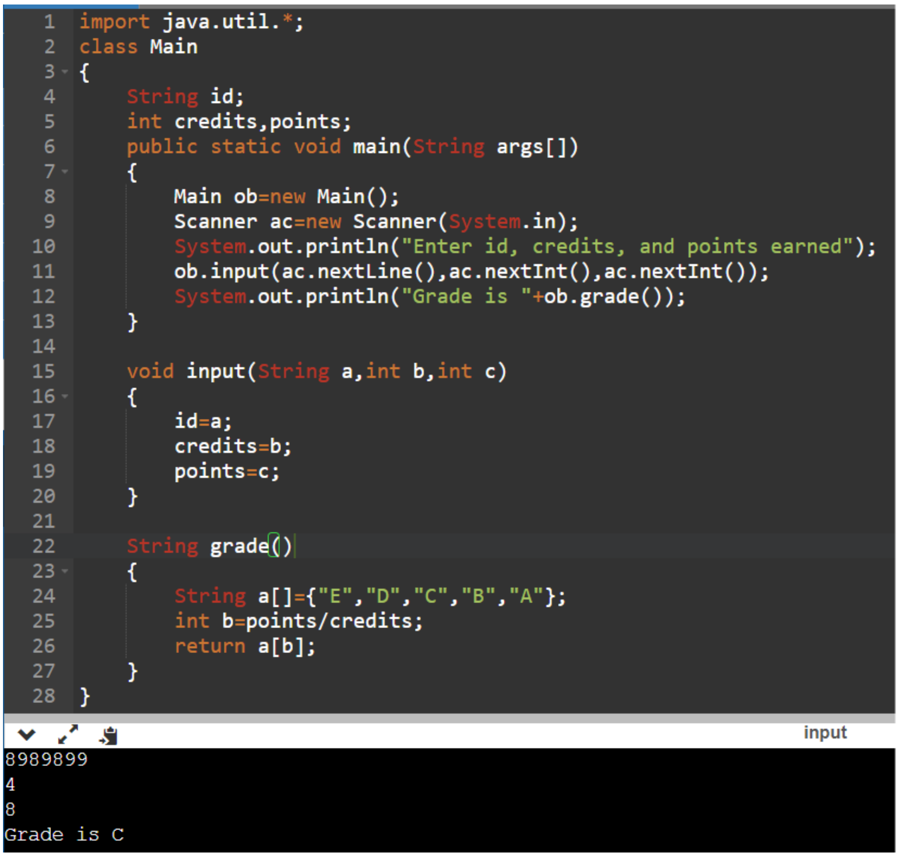


```java
import java.util.*;
class Main
{
    public static void main(String args[])
    {
        Scanner ac=new Scanner(System.in);
        Main ob=new Main();
        System.out.println("Enter the name, The cost of material, numbers of hours work required, travel time");
        String a=ac.nextLine();
        int x=ac.nextInt();
        int y=ac.nextInt();
        int z=ac.nextInt();
        System.out.println("cost will be Rupees "+ob.cost(x,y,z));
    }

    int cost(int x,int y,int z)
    {
        return x+y*32+z*12;
    }
}
```

```
Enter the name, The cost of material, numbers of hours work required, travel time
Jaswanth
5000
3
2
cost will be Rupees 5120


...Program finished with exit code 0
Press ENTER to exit console.
```

### Question 5a

Create a class named Student that has fields for an ID number, number of credit hours earned, and number of points earned. (For example, many schools compute grade point averages based on a scale of 4, so a three-credit-hour class in which a student earns an A is worth 12 points.) Include methods to assign values to all fields. A Student also has a field for grade point average. Include a method to compute the grade point average field by dividing points by credit hours earned. Write methods to display the values in each Student field. Save this class as Student.java.



```java
import java.util.*;
class Main
{
    String id;
    int credits,points;
    public static void main(String args[])
    {
        Main ob=new Main();
        Scanner ac=new Scanner(System.in);
        System.out.println("Enter id, credits, and points earned");
        ob.input(ac.nextLine(),ac.nextInt(),ac.nextInt());
        System.out.println("Grade is "+ob.grade());
    }

    void input(String a,int b,int c)
    {
        id=a;
        credits=b;
        points=c;
    }

    String grade()
    {
        String a[]={"E","D","C","B","A"};
        int b=points/credits;
        return a[b];
    }
}
```

```
8989899
4
8
Grade is C
```

Question 5b

**Write a class named ShowStudent that instantiates a Student object from the class you created and assign values to its fields. Compute the Student grade point average, and then display all the values associated with the Student. Save the application as ShowStudent.java.**

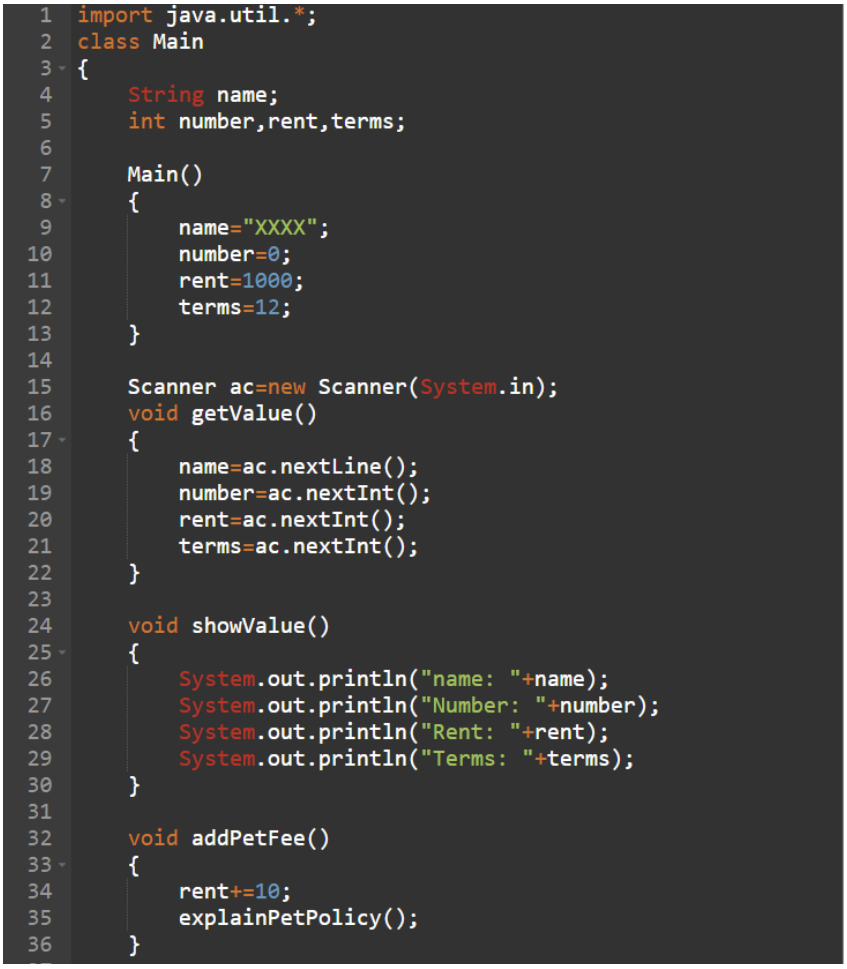
**Question 5c**

**Create a constructor for the Student class you created. The constructor should initialize each Student's ID number to 9999, his or her points earned to 12, and credit hours to 3 (resulting in a grade point average of 4.0). Write a program that demonstrates that the constructor works by instantiating an object and displaying the initial values. Save the application as ShowStudent2.java.**

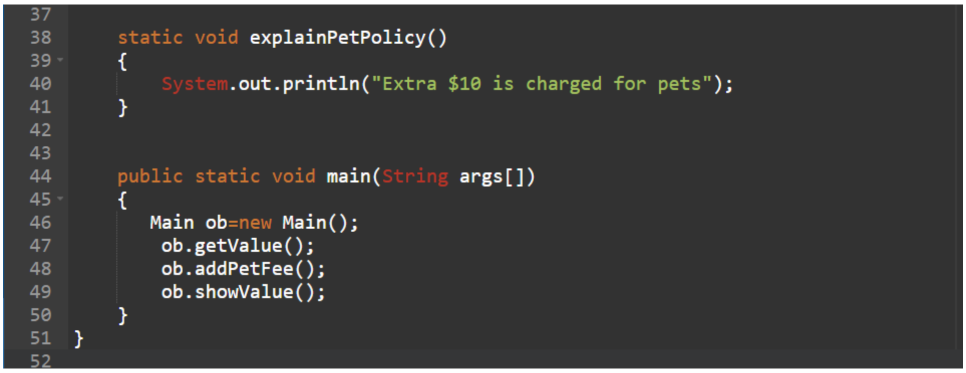
Question 6a

**Create a class named Lease with fields that hold an apartment tenant's name, apartment number, monthly rent amount, and term of the lease in months.**

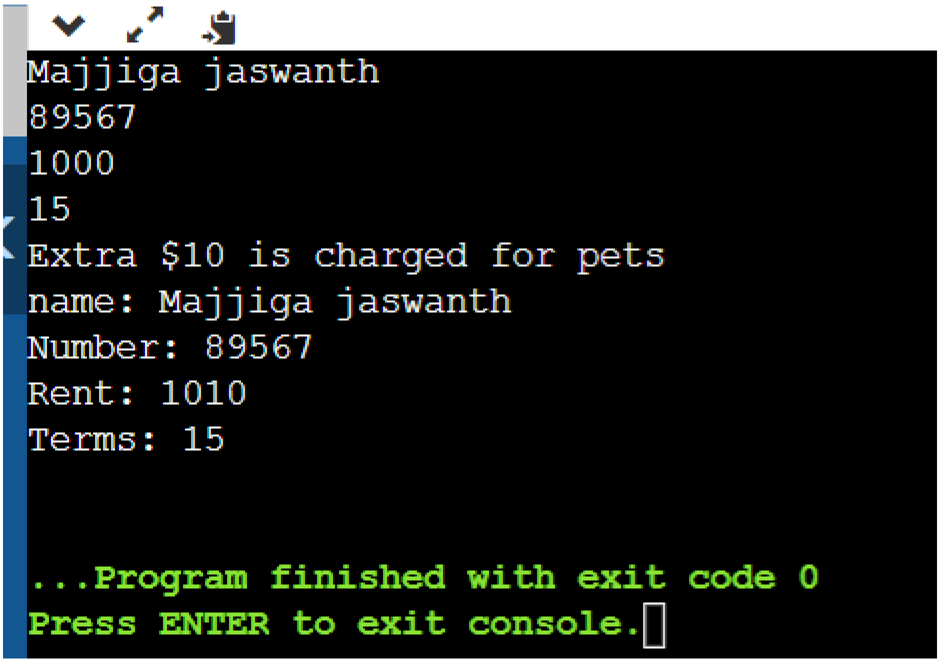
Include a constructor that initializes the name to “XXX”, the apartment number to 0, the rent to 1000, and the term to 12. Also include methods to get and set each of the fields. Include a nonstatic method named addPetFee() that adds $10 to the monthly rent value and calls a static method named explainPetPolicy() that explains the pet fee. Save the class as Lease.java.



```java
import java.util.*;
class Main
{
    String name;
    int number,rent,terms;

    Main()
    {
        name="XXXX";
        number=0;
        rent=1000;
        terms=12;
    }

    Scanner ac=new Scanner(System.in);
    void getValue()
    {
        name=ac.nextLine();
        number=ac.nextInt();
        rent=ac.nextInt();
        terms=ac.nextInt();
    }

    void showValue()
    {
        System.out.println("name:  "+name);
        System.out.println("Number:  "+number);
        System.out.println("Rent:  "+rent);
        System.out.println("Terms:  "+terms);
    }

    void addPetFee()
    {
        rent+=10;
        explainPetPolicy();
    }
```



```java

    static void explainPetPolicy()
    {
        System.out.println("Extra $10 is charged for pets");
    }


    public static void main(String args[])
    {
       Main ob=new Main();
        ob.getValue();
        ob.addPetFee();
        ob.showValue();
    }
}

```



```
Majjiga jaswanth
89567
1000
15
Extra $10 is charged for pets
name: Majjiga jaswanth
Number: 89567
Rent: 1010
Terms: 15


...Program finished with exit code 0
Press ENTER to exit console.
```

Question 6b

Create a class named TestLease whose main() method declares four Lease objects. Call a getData() method three times. Within the method, prompt a user for values for each field for a Lease, and return a Lease object to the main() method where it is assigned to one of main()'s Lease objects. Do not prompt the user for values for the fourth Lease object, but let it continue to hold the default values. Then, in main(), pass one of the Lease objects to a showValues() method that displays the data. Then call the addPetFee() method using the passed Lease object and confirm that the fee explanation statement is displayed. Next, call the showValues() method for the Lease object again and confirm that the pet fee has been added to the rent. Finally, call the showValues() method with each of the other three objects; confirm that two hold the values you supplied as input and one holds the constructor default values. Save the application as TestLease.java.