Java Programms

55. Write a program to demonstrate the use of final keyword.

// final keyword is used to assign the final value to a variable; thus, the compiler wouldn’t take any other value for the particular variable.

class Main

{

public static void main(String[] args) {

final int age=32;

age=45;

System.out.println("age is: "+age);

}

}

class A extends Main{

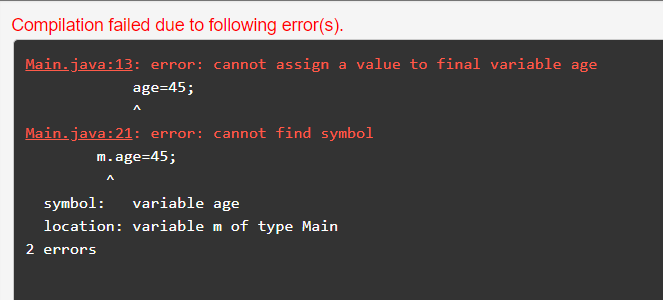
public static void main(String[] args){

Main m = new Main();

m.age=45;

}

}



56. Create a final method to demonstrate its working and override a method with final keyword.

class FinalDemo

{

public void display()

{

System.out.println("This is a final method!");

}

}

class Main extends FinalDemo

{

public final void display()

{

System.out.println("The final method is overridden!");

}

public static void main(String[] args)

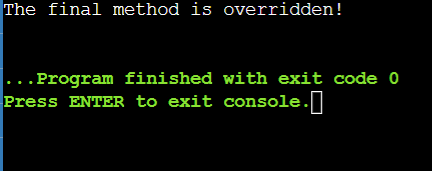
{

Main m = new Main();

m.display();

}

}



57. To demonstrate the use of ‘this’ keyword; also demonstrate constructor overriding.

class Main{

int sum;

Main(){

this(5,2);

}

Main(int a1, int a2){

this.sum=a1+a2;

}

void display()

{

System.out.println("Sum is: "+sum);

}

public static void main(String[] args)

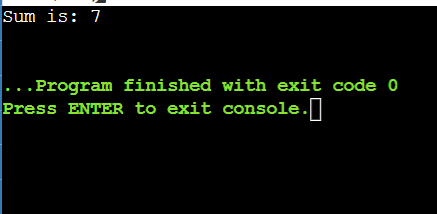
{

Main m = new Main();

m.display();

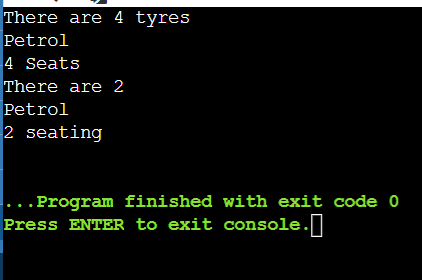
}

}



58. Create a class called Automobile – create two child classes and perform method overriding.

class Automobile  
{  
public void stop()  
{  
   System.out.println("Engine stops running,");  
}  
public void start()  
{  
   System.out.println("Engine starts runnimg");  
}  
public void seating()  
{  
}  
}  
class Car extends Automobile{  
    public void tyre()  
    {  
        System.out.println("There are 4 tyres");  
    }  
    public void fuel()  
    {  
        System.out.println("Petrol");  
    }  
    public void fuel(String fuel)  
    {  
        System.out.println("Also on Diesel");  
    }  
    public void seating()  
    {  
        System.out.println("4 Seats");  
    }  
}  
class Bike extends Automobile  
{  
    public void tyre()  
    {  
        System.out.println("There are 2");  
    }  
    public void fuel()  
    {  
        System.out.println("Petrol");  
    }  
    public void seating()  
    {  
        System.out.println("2 seating");  
    }  
}  
public class Main{  
    public static void main(String [] args)  
    {  
        Car c=new Car();  
        c.tyre();  
        c.fuel();  
        c.seating();  
        Bike b= new Bike();  
        b.tyre();  
        b.fuel();  
        b.seating();  
    }  
  
}



59. Create a interface and extend it as a class.

interface solution{

public void Hello();

public void Welcome();

public void Name(String s);

}

public class A implements solution{

public void Hello()

{

java.lang.System.out.println("Hello, World!");

}

public void Welcome()

{

java.lang.System.out.println("Welcome to Edureka!");

}

public void Name(String s)

{

java.lang.System.out.println("My name is Harshita!");

}

public static void main(String[] args)

{

A a = new A();

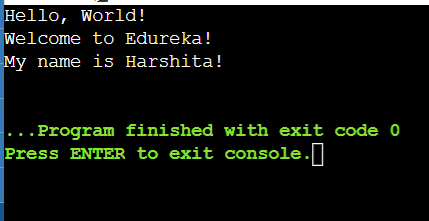
a.Hello();

a.Welcome();

a.Name("A");

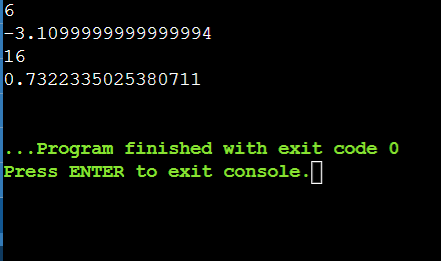
}

}



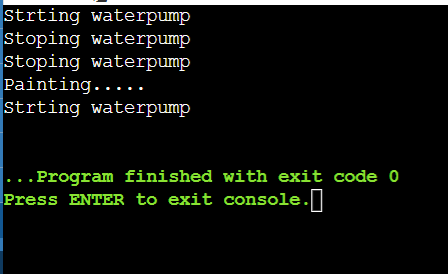
60.  Create an interface named calculator, add, subtract, multiply, divide - signature. create class User to implement.

interface Calci{  
    public void Add(int a, int b);  
    public void Minus(double a, double b);  
    public void Multiply(int a, int b);  
    public void Divide(double a, double b);  
}  
  
public class B implements Calci{  
    public void Add(int a, int b)  
    {  
        int c=a+b;  
        java.lang.System.out.println(c);  
    }  
     
    public void Minus(double a, double b)  
    {  
       double c=a-b;  
        java.lang.System.out.println(c);  
    }  
     
    public void Multiply(int a, int b)  
     
    {  
        int c=a\*b;  
        java.lang.System.out.println(c);  
    }  
     
    public void Divide(double a, double b)  
    {  
       double c=a/b;  
       java.lang.System.out.println(c);  
    }  
    public static void main(String[] args)  
    {  
        B a = new B();  
        a.Add(2,4);  
        a.Minus(3.66 , 6.77);  
        a.Multiply(4,4);  
        a.Divide(5.77, 7.88);  
    }  
}



61. Create an interface named machine and create suitable methods.

interface Machine{  
    public void start();  
    public void stop();  
}  
  
class waterpump implements Machine  
{  
    public void start()  
    {  
        System.out.println("Strting waterpump");  
    }  
     
    public void stop()  
    {  
        System.out.println("Stoping waterpump");  
    }  
}  
  
public class Testing  
{  
    public static void main(String[] args)  
    {  
        Machine m= new waterpump();  
        m.start();  
        m.stop();  
        paint(new waterpump());  
    }  
     
    public static void paint(Machine m)  
    {  
        m.stop();  
        System.out.println("Painting.....");  
        m.start();  
    }  
}



62.