23CSE111

OBJECT ORIENTED PROGRAMMING

LAB REPORT

Department of Computer Science Engineering

Amrita School of Computing

Amrita Vishwa Vidyapeetham, Amaravati Campus

Verified By

Name: A.Krishna Hasini

ROLL NO:24013…..

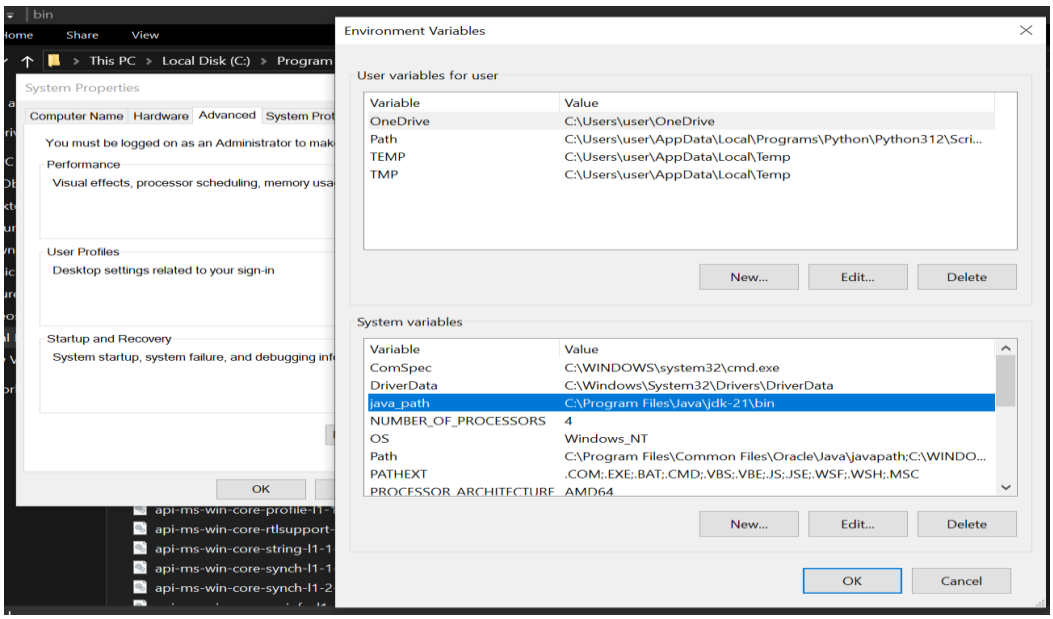
WEEK-1

TASK-1

AIM: Installing of JDK (Java Development Kit)

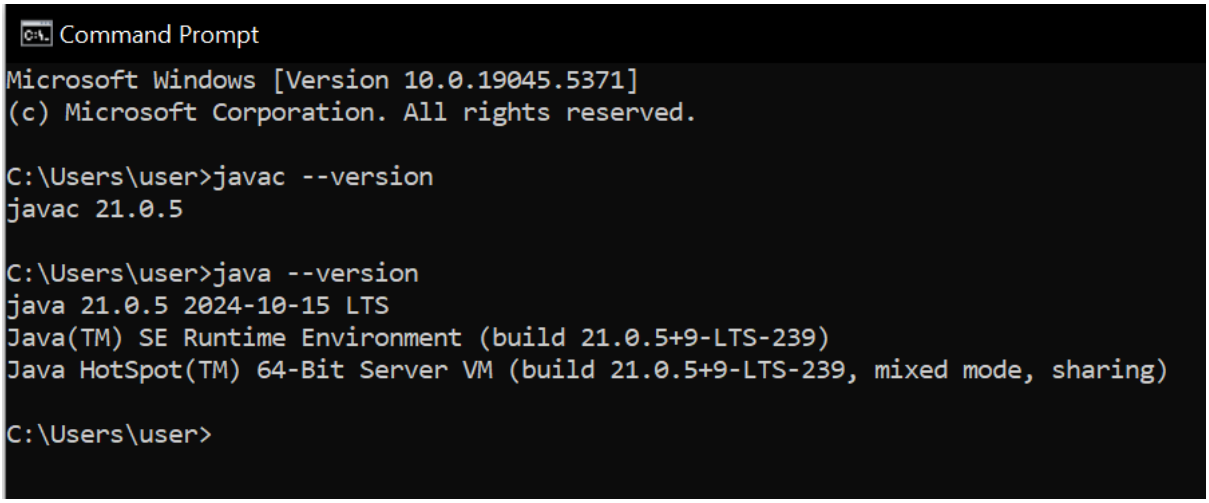
PROCEDURE:

1. **Download JDK:**
   - Go to the Oracle JDK download page in your web browser and click on JDK-21 version which is Long term support (LTS) version.
   - Click on the download link for your operating system (Windows, macOS, or Linux).
2. **Install JDK:**
   - Once downloaded, run the installer.
   - Follow the instructions and keep clicking "Next" until it's done.
3. **Set Environment Variables (Windows):**
   - Open file explorer, then right click on This PC next select on properties then it will take you to the settings app then click on advanced system settings and then
     click on **Environment Variables**.
   - Click **New** under **System Variables**:
     - **Set Variable name as:** java_home
     - **Variable value:** The folder address where JDK is installed (like C:\Program Files\Java\jdk-21\bin)
     - Find Path under **System Variables**, click **Edit**, and add the path of the jdk-21(C:\Program Files\Java\jdk-21\bin)



Checking of JDK Version:

1. **Open Command Prompt:**
    - Press win+R, type cmd, and press Enter.
2. **Check Version:**
    - Type java --version and press Enter.
    - Type javac --version and press Enter.



TASK-2

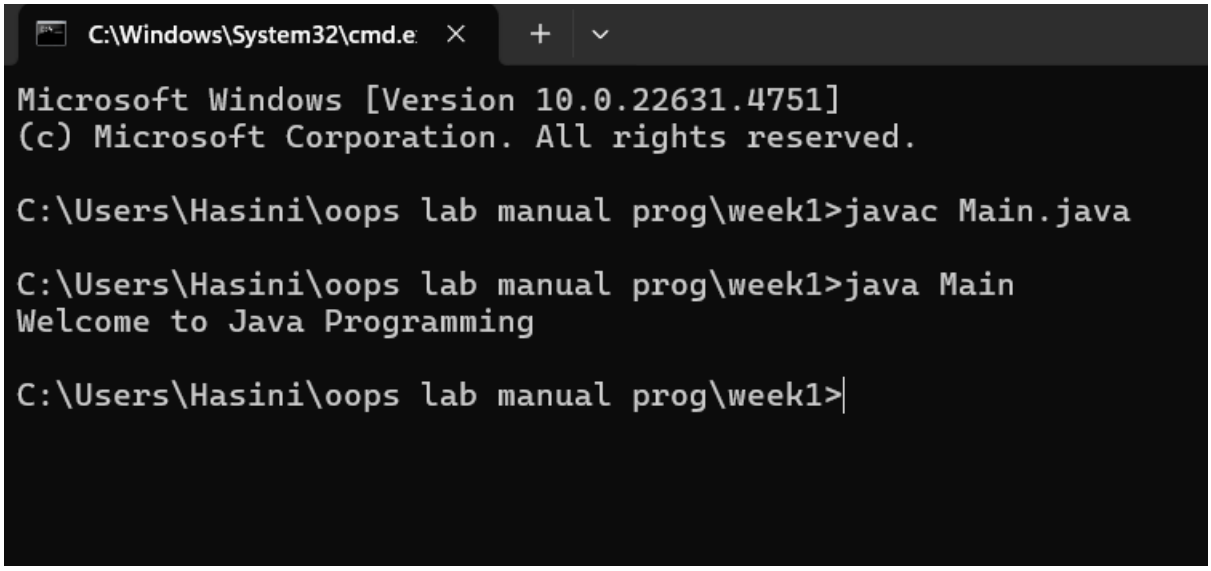
AIM: Simple Java Program for printing a message "Welcome to Java Programming".

Write your code in Notepad and execute in cmd prompt

CODE:

```
class Main
{
   public static void main(String[] args)
   {
      System.out.println("Welcome to Java Programming");
   }
}
```

Output:



ERRORS:

Sno.	Error message	Error rectification
1.	error: ';' expected	Adding a semi-colon at the last

	System.out.println("Welcome to Java Programming")	**System.out.println("Welcome to Java Programming");**
2.	**error: reached end of file while parsing** **}**	**Placing a curly bracket at the end of file, to close the class**

IMP POINTS:

- **Every Java statement must end with a semicolon (;).**
- **If using a filename, it should match the class name (Main.java for class Main). Java is case-sensitive (Main ≠ main)**

TASK-3

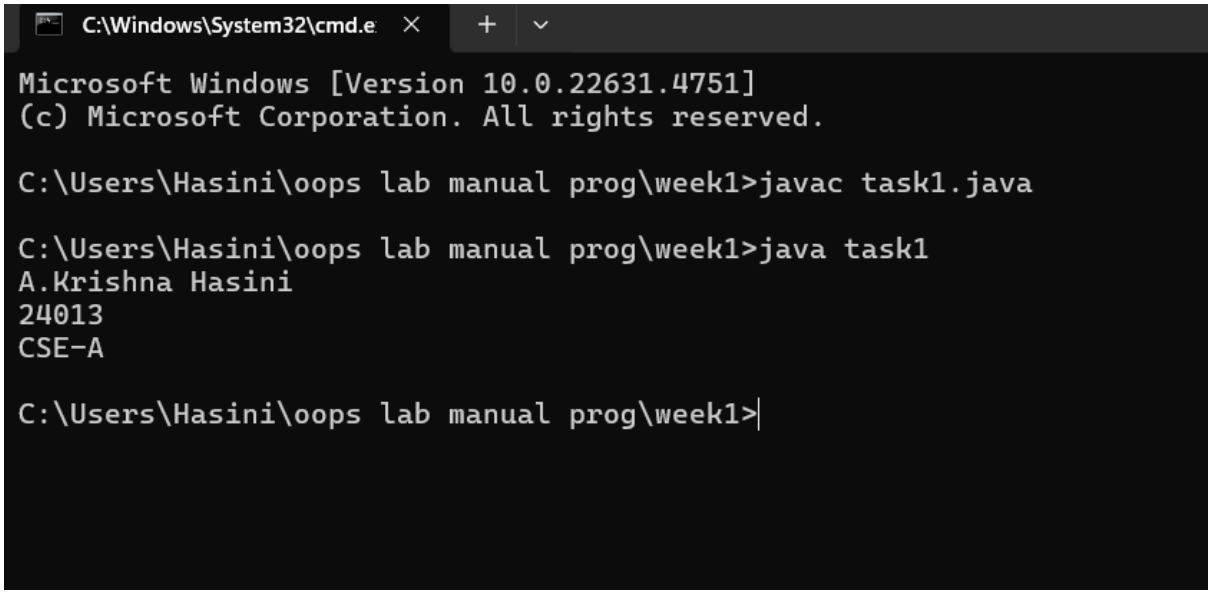
AIM: Simple Java Program for printing Name, Class, Roll No, of a Student

Write your code in Notepad and execute in cmd prompt

CODE:

```
class Main
{
   public static void main(String[] args)
   {
      System.out.println("Name: A.Krishna Hasini");
      System.out.println("Section: CSE-A");
      System.out.println("Roll No : 24013")
}
}
```

Output:



ERRORS:

Sno.	Error message	Error rectification
1.	**error: reached end of file while parsing** **}**	**Placing a curly bracket at the end of file, to close the class**

IMP POINTS:

- Every Java statement must end with a semicolon (;).
- If using a filename, it should match the class name (Main.java for class Main). Java is case-sensitive (Main ≠ main)
- Every Java program must have at least one class

WEEK 2

Program 1

Write a java program to calculate the area of a rectangle.

Code:

```
import java.util.Scanner;
class rectangle{
        public static void main(String[] args){
        //to calculate the area of rectangle
        Scanner input=new Scanner(System.in);
        System.out.print("enter the length of a rectangle:");
        int ln=input.nextInt();
        System.out.print("enter the breadth of a rectangle:");
        int bd=input.nextInt();
        int ar=ln*bd;
        System.out.println("area of a rectangle is:"+ar);
        }
}
```

Output:

```
D:\java>javac rectangle.java

D:\java>java rectangle
enter the length of a rectangle:3
enter the breadth of a rectangle:9
area of a rectangle is:27
```

Error:

Sno.	Error message	Error rectification
1.	error: incompatible types: Scanner cannot be converted to System System input= new Scanner(System.in);	Change System into Scanner

IMP POINTS:

1. **import java.util.Scanner; -** To accept input from user, Scanner class under util package has to be imported.
2. **Scanner input=new Scanner(System.in); -** Used to create a Scanner object

3. **int ln=input.nextInt(); -** Used to read the integer data type stored under the object create
4. **System.out.println(" ");** - It is used to print string inside the quotes. After printing, the cursor moves to the beginning of the next line**.**

Program 2

Write a java program to convert temperature from Celsius to Fahrenheit and vica-versa.

Code:

```
import java.util.Scanner;
class temperature{
        public static void main(String[] args){
        Scanner input=new Scanner(System.in);
        System.out.print("enter the temp in celcius:");
        double tp=input.nextDouble();
        double fh=(tp*9/5)+32;
        System.out.println("the temperature in Fahrenheit is:"+fh);
        System.out.print("enter the temp in Fahrenheit:");
        double tp1=input.nextDouble();
        double cl=(tp1-32)*5/9;
        System.out.println("the temperature in celcius is:"+cl);
        }
}
```

Output:

```
D:\java>javac temperature.java

D:\java>java temperature
enter the temp in celcius:100
the temperature in Fahrenheit is:212.0
enter the temp in Fahrenheit:50
the temperature in celcius is:10.0
```

Error:

Sno.	Error message	Error rectification
1.	error: cannot find symbol fh=(tp*9/5)+32;	Declare the variable: double fh=(tp*9/5)+32;
2.	error: ';' expected System.out.print("Enter the temp in Farenheit:")	Add a semicolon at the end of the statement System.out.print("Enter the temp in Farenheit:");

IMP POINTS:

1. **import java.util.Scanner;** - To accept input from user, Scanner class under util package has to be imported.

2. **Scanner input=new Scanner(System.in); -** Used to create a Scanner object
3. **double fh=input.nextDouble(); -** Used to read double data type stored under the object created
4. **System.out.println(" "); -** It is used to print string inside the quotes. After printing, the cursor moves to the beginning of the next line.

Program 3:

Write a java program to calculate the simple interest.

Code:

```
import java.util.Scanner;
class interest{
        public static void main(String[] args){
        Scanner input=new Scanner(System.in);
        System.out.print("enter the principal amount:");
        double p=input.nextDouble();
        System.out.print("enter the rate:");
        double r=input.nextDouble();
        System.out.print("enter the time period:");
        double t=input.nextDouble();
        double intr=(p*t*r)/100;
        System.out.println("simple interest:"+intr);
        }
}
```

Output:

```
D:\java>javac interest.java

D:\java>java interest
enter the principal amount:100
enter the rate:5
enter the time period:5
simple interest:25.0
```

Error:

Sno.	Error message	Error rectification
1.	error: ';' expected double intr=(p*r*t)/100	Add a semicolon at the end of the statement double intr=(p*r*t)/100;
2.	error: cannot find symbol double intr=(p*r*t)/100; symbol: variable p location: class interest	Create a reader object double p=input.nextDouble();

IMP POINTS:

1. import java.util.Scanner; - To accept input from user, Scanner class under util package has to be imported.
2. Scanner input=new Scanner(System.in); - Used to create a Scanner object
3. double p=input.nextDouble(); - Used to read double data type stored under the object created
4. System.out.println(" "); - It is used to print string inside the quotes. After printing, the cursor moves to the beginning of the next line.

Program 4

Write a java program to find the largest of three numbers, using ternary operator.

Code:

```
import java.util.Scanner;
class ternary{
        public static void main(String[] args){
        //use ternary operator
        Scanner input=new Scanner(System.in);
        System.out.print("enter first number:");
        int a=input.nextInt();
        System.out.print("enter second number:");
        int b=input.nextInt();
        System.out.print("enter third number:");
        int c=input.nextInt();
        int result=(a>b)? ((a>c)? a:c) : ((b>c)? b:c);
        System.out.println("largest number is:"+result);
        }
}
```

Output:

```
D:\java>javac ternary.java

D:\java>java ternary
enter first number:5
enter second number:7
enter third number:9
largest number is:9
```

Error:

Sno.	Error message	Error rectification
1.	error: ';' expected int result=(a>b) ((a>c)? a:c) : ((b>c)? b:c); error: not a statement int result=(a>b) ((a>c)? a:c) : ((b>c)? b:c);	Add a '?' int result=(a>b)? ((a>c)? a:c) : ((b>c)? b:c);

2.	error: ';' expected int result=(a>b)? ((a>c)? a:c) : ((b>c)? b:c)	Add a ';' int result=(a>b)? ((a>c)? a:c) : ((b>c)? b:c);

IMP POINTS:

1. import java.util.Scanner; - To accept input from user, Scanner class under util package has to be imported.
2. Scanner input=new Scanner(System.in); - Used to create a Scanner object
3. int a=input.nextInt (); - Used to read integer data type stored under the object created
4. int result=(a>b)? ((a>c)? a:c) : ((b>c)? b:c); - Nested Ternary operator is used here.

Syntax for ternary operator is- condition? expression 1: expression 2; , whose answer is stored in a variable and then used.

Program 5

Write a java program to find the factorial of a number.

Code:

```
import java.util.Scanner;
class factorial{
        public static void main(String[] args){
        //to calculate the factorial of a number
        Scanner input=new Scanner(System.in);
        System.out.print("enter a number:");
        int n=input.nextInt();
        int fact=1;
        for (int i=1; n>=i;--n){
                fact*=n;
        }
        System.out.println("factorial:"+fact);
        }
}
```

Output:

```
D:\java>javac factorial.java

D:\java>java factorial
enter a number:5
factorial:120
```

Error:

Sno.	Error message	Error rectification
1.	error: ';' expected	Add a ";"

	fact*=n	fact*=n;

IMP POINTS:

1. for (int i=1; n>=i;--n){ } - For loop syntax: for(initial expression; test expression; update expression){} The loop is executed, until the test expression evaluates to be false.

WEEK- 3

Program 1

Create a java program with the following instructions.

I. Create a class with name car
II. Create four attributes named car color,car brand,fuel type and milage.
III. Create three methods named start(),stop(),service().
IV. Create three objects named car1,car2,car3.

Code:

```
class car
{
	public String car_color;
	public String car_brand;
	public String fuel_type;
	public float mileage;
	public void start()
	{
		System.out.println("Car starts");
	}
	public void stop()
	{
		System.out.println("Car stops");
	}
	public void service()
	{
		System.out.println("Car service");
	}
	public static void main(String [] args){
			// object 1 creation
			car car1= new car();
			car1.car_color="White";
			car1.car_brand="Toyota";
			car1.fuel_type="Diesel";
			car1.mileage=13.8F;

			//calling methods for object 1
			car1.start();
			car1.stop();
			car1.service();
			System.out.println("Color of the car1 is "+car1.car_color);
			System.out.println("Brand of the car1 is "+car1.car_brand);
			System.out.println("Fuel type of the car1 is "+car1.fuel_type);
			System.out.println("Mileage of the car1 is"+car1.mileage);
```

```
        // object 2 creation
        car car2= new car();
        car2.car_color="Black";
        car2.car_brand="Mahindra";
        car2.fuel_type="Diesel";
        car2.mileage=17.3F;

        // calling methods for object 2
    car2.start();
        car2.stop();
        car2.service();
        System.out.println("Color of the car2 is "+car2.car_color);
        System.out.println("Brand of the car2 is "+car2.car_brand);
        System.out.println("Fuel type of the car2 is "+car2.fuel_type);
        System.out.println("Mileage of the car2 is"+car2.mileage);


        //object 3 creation
        car car3= new car();
        car3.car_color="Brown";
        car3.car_brand="Ford";
        car3.fuel_type="Diesel";
        car3.mileage=15.5F;

        // calling methods for object 3
        car3.start();
        car3.stop();
        car3.service();
        System.out.println("Color of the car3 is "+car3.car_color);
        System.out.println("Brand of the car3 is "+car3.car_brand);
        System.out.println("Fuel type of the car3 is "+car3.fuel_type);
        System.out.println("Mileage of the car3 is"+car3.mileage)  }
}
```

OUTPUT:

```
C:\Users\Hasini\oops lab manual prog\week1>javac car_oop.java

C:\Users\Hasini\oops lab manual prog\week1>java car_oop
Car starts
Car stops
Car service
Color of the car1 is White
Brand of the car1 is Toyota
Fuel type of the car1 is Diesel
Mileage of the car1 is13.8
Car starts
Car stops
Car service
Color of the car2 is Black
Brand of the car2 is Mahindra
Fuel type of the car2 is Diesel
Mileage of the car2 is17.3
Car starts
Car stops
Car service
Color of the car3 is Brown
Brand of the car3 is Ford
Fuel type of the car3 is Diesel
Mileage of the car3 is15.5

C:\Users\Hasini\oops lab manual prog\week1>
```

ERROR:

Sno.	Error message	Error rectification
1.	**error: incompatible types: Scanner cannot be converted to System** **System input= new Scanner(System.in);**	**Change System into Scanner**

IMP POINTS:

- A **class** (Car) acts as a blueprint.
- An **object** is an instance of a class (new Car(...)).
- Use **private** variables and provide **public** methods to access or modify them.
- A **constructor** initializes an object with specific values when it is created.

Program 2

Create a java program with the following instructions.

I. Create a class with named bank account
II. Create two methods named withdraw(),deposit().

Code:

```
import java.util.Scanner;

class BankAccount {
    String name;
    int AccountNum;
    float currentAmount;

    public BankAccount(String name, int AccountNum, float
currentAmount) {
        this.name = name;
        this.AccountNum = AccountNum;
        this.currentAmount = currentAmount;
    }

    public void deposit() {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter amount to be Deposited: ");
        float deposit = input.nextFloat();
        currentAmount += deposit;
        System.out.println("Existing Balance now is: " +
currentAmount);
    }
```

```
    public void withdrawal() {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter amount to be withdrawn: ");
        float withdrawal = input.nextFloat();

        if (currentAmount < withdrawal) {
            System.out.println("Insufficient Funds in account");
        } else {
            currentAmount -= withdrawal;
            System.out.println("Remaining Balance: " + currentAmount);
        }
    }

    public static void main(String[] args) {
        BankAccount BA = new BankAccount("Veena", 123456, 100000);
BA.withdrawal();
 BankAccount BA1=new BankAccount(“Hasini",9876543,200000);
BA1.deposit();

    }

}
```

OUTPUT:

```
C:\Users\Hasini\oops lab manual prog\week1>javac BankAccount.java

C:\Users\Hasini\oops lab manual prog\week1>java BankAccount
Enter amount to be withdrawn:
6000
Remaining Balance: 94000.0
Enter amount to be Deposited:
10000
Existing Balance now is: 210000.0

C:\Users\Hasini\oops lab manual prog\week1>
```

ERROR:

Sno.	Error message	Error rectification
1.	**error: incompatible types: Scanner cannot be converted to System** **System input= new Scanner(System.in);**	**Change System into Scanner**

IMP POINTS:

- A **class** (BankAccount) acts as a blueprint.
- An **object** is an instance of a class (new BankAccount(...)).

- Use **private** variables and provide **public** methods to access or modify them.
- A **constructor** initializes an object with specific values when it is created.

WEEK-4

Program 1:

1) Write a java program with class named "book" the class should contain various attributes such as "title","author","year_publication".It should also contain a constructor with parameters which initializes "title","author" and "year_publication".

- **Create a method which displays the details of the book i.e."title","author" and "year_Publication".**
- **Display the details of two books by creating two objects.**

Code:

```
public class book_oop {
        String title,Author;
        int year;
        book_oop(String title,String Author,int year){
                this.title=title;
                this.Author=Author;
                this.year=year;
        }
        public void display() {
                System.out.println(title+" "+Author+" "+year);
        }
        public static void main(String[] args) {
                book_oop book1=new book_oop("wings","keyfer",2019);
                book1.display();
                book_oop book2=new book_oop("Bravery","James",2004);
                book2.display();
        }

}
```

OUTPUT:

```
C:\Users\Hasini\eclipse-workspace\oops_prob\src\oops_prob>javac book_oop.java

C:\Users\Hasini\eclipse-workspace\oops_prob\src\oops_prob>java book_oop
wings keyfer 2019
Bravery James 2004

C:\Users\Hasini\eclipse-workspace\oops_prob\src\oops_prob>
```

ERRORS:

Sno.	Error message	Error rectification
1.	error: could not load main class;	Keep class name and file name same.

IMP POINTS:

- **Keep same name for class and file.**
- **Keep class name for constructor.**
- **To print various variables we should keep '+' between them, not ','.**

Program-2:To create a java program with class named "My_class" with a static variable "count" of "int" type,initialized to 0 and a constant variable "pi" of type "double" initialized to 3.1415 as attributes of that class.Define a constructor for "My_class" that increments the count variable each time an object of "My_class" is created finally print the final values of "count" and "pi" values.

```
class my_class {
        static int count=0;
        final double pi=3.1415;
        my_class(){
                count++;
        }
        public void display() {
                System.out.println(count+" "+pi);
        }
        public static void main(String[] args) {
                my_class obj1=new my_class();
                obj1.display();
                my_class obj2=new my_class();
                obj2.display();
        }
}
```

OUTPUT:

```
C:\Users\Hasini\eclipse-workspace\oops_prob\src\oops_prob>javac my_class.java

C:\Users\Hasini\eclipse-workspace\oops_prob\src\oops_prob>java my_class
1 3.1415
2 3.1415

C:\Users\Hasini\eclipse-workspace\oops_prob\src\oops_prob>
```

Errors:

Sno.	Error message	Error rectification
1.	Error: Static;	Static
2.	Error:float pi	double pi

IMP POINTS:

- **Use ‘static’ keyword to access class level variables.**
- **Use ‘final’ keyword for constant variables.**