Graded Exercise

Object Oriented Programming and Design with Java

–

20CS43P

IV Semester

Computer Science

Certificate

Name: **Class: IV Semester**

Register Number:

Institution:

This is certified to be the bonafide work of the student in the **ObjectOriented Programming and Design with Java – 20CS43P**Laboratory during the academic year 20 – 20

Course Coordinator

Examiner Signature

1. ______________________

2. ______________________

Name: **Class: IV Semester**
Register Number: **Year: 2021 - 2022**

List of Experiments

SL NO	Name of Experiment	Marks	Remarks
01			
02			
03			
04			
05			
06			
07			
08			
09			
10			
11			
12			
13			
14			

15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			

Average Marks: ___ / 10

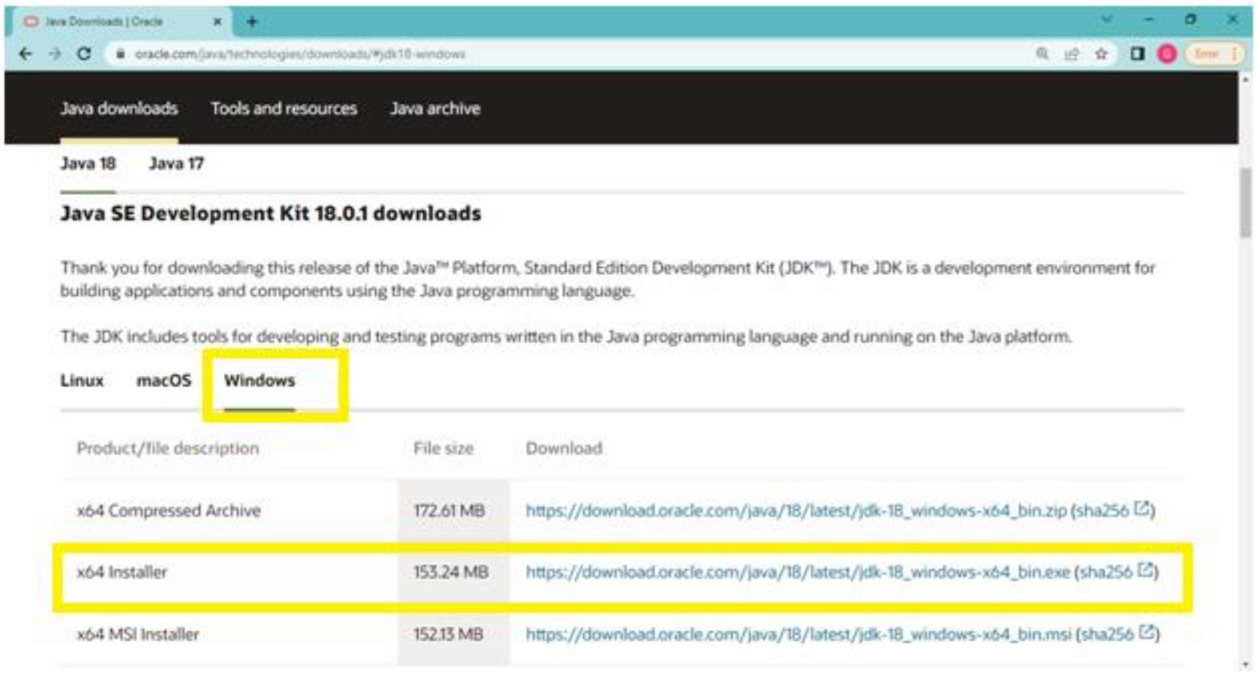
Signature of the student **Signature of the Course Coordinator**

Exercise 01:
Install and Setup java environment

Java environment setup

Steps for setting the environment in Windows operation system are as follows:

Step 1: Java18 JDK is available at “https://www.oracle.com/java/technologies/downloads/#jdk18-windows”. Go to the Windows tab and Click the second last link for download as highlighted below.



Java SE Development Kit 18.0.1 downloads

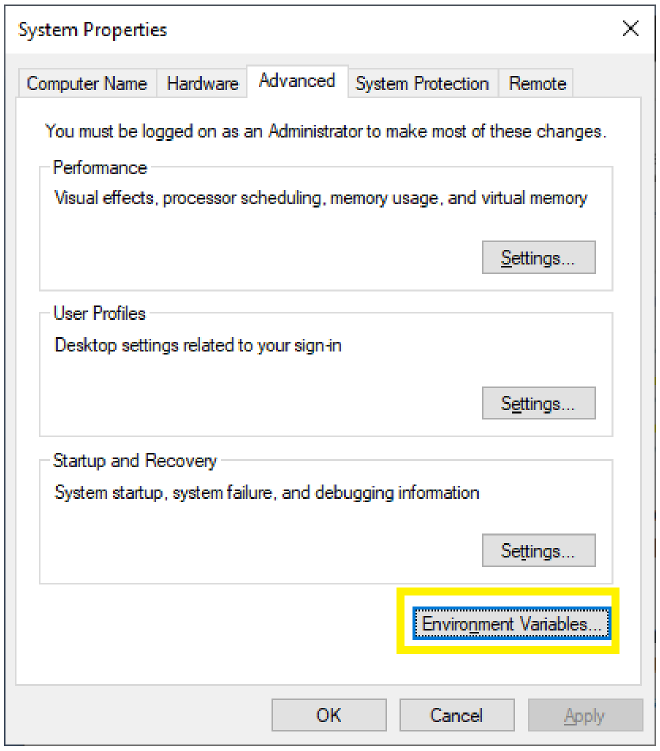
Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications and components using the Java programming language.

The JDK includes tools for developing and testing programs written in the Java programming language and running on the Java platform.

Product/file description	File size	Download
x64 Compressed Archive	172.61 MB	https://download.oracle.com/java/18/latest/jdk-18_windows-x64_bin.zip (sha256)
x64 Installer	153.24 MB	https://download.oracle.com/java/18/latest/jdk-18_windows-x64_bin.exe (sha256)
x64 MSI Installer	152.13 MB	https://download.oracle.com/java/18/latest/jdk-18_windows-x64_bin.msi (sha256)

Step 2: After download, run the .exe file and follow the instructions to install Java on your machine. Once you installed Java on your machine, you have to set up the environment variable.

Step 3: Go to **Start** and search for **“Environment Variables”.** Click on **“Edit the system Environment Variable**s”. Under the Advanced System Setting option click on **Environment Variables** as highlighted below.


System Properties

Computer Name | Hardware | Advanced | System Protection | Remote

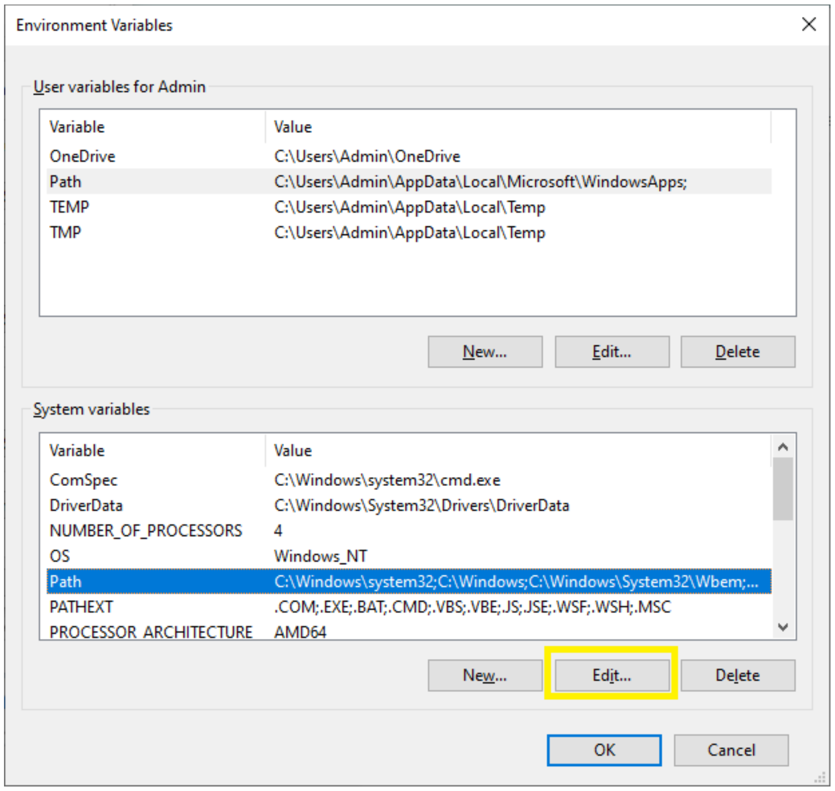
You must be logged on as an Administrator to make most of these changes.

Performance
Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles
Desktop settings related to your sign-in

Startup and Recovery
System startup, system failure, and debugging information

Step 4: Now, you have to alter the "Path" variable under System variables so that it also contains the path to the Java environment. Select the "Path" variable and click on the Edit button as highlighted below.


Environment Variables

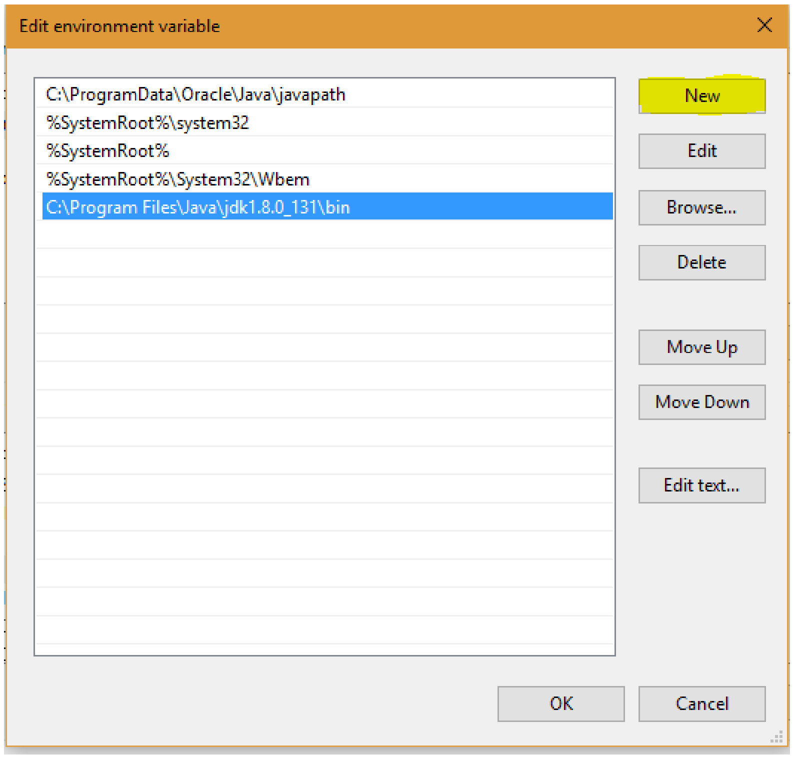
User variables for Admin

Variable	Value
OneDrive	C:\Users\Admin\OneDrive
Path	C:\Users\Admin\AppData\Local\Microsoft\WindowsApps;
TEMP	C:\Users\Admin\AppData\Local\Temp
TMP	C:\Users\Admin\AppData\Local\Temp

System variables

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

Step 5: You will see a list of different paths, click on the New button, and then add the path where java is installed. By default, java is installed in “C:\Program Files\Java\jdk\bin” folder OR “C:\Program Files(x86)\Java\jdk\bin”. In case, you have installed java at any other location, then add that path.


Edit environment variable

C:\ProgramData\Oracle\Java\javapath
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
C:\Program Files\Java\jdk1.8.0_131\bin

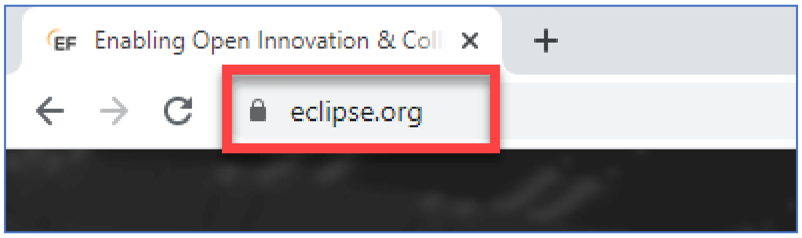
Step 6: Click on OK, Save the settings. Now to check whether the installation is done correctly, open the command prompt and type javac -version. You will see that java is running on your machine.

Exercise 02:
Install java editor (Eclipse for Enterprise Java) and configure workspace.

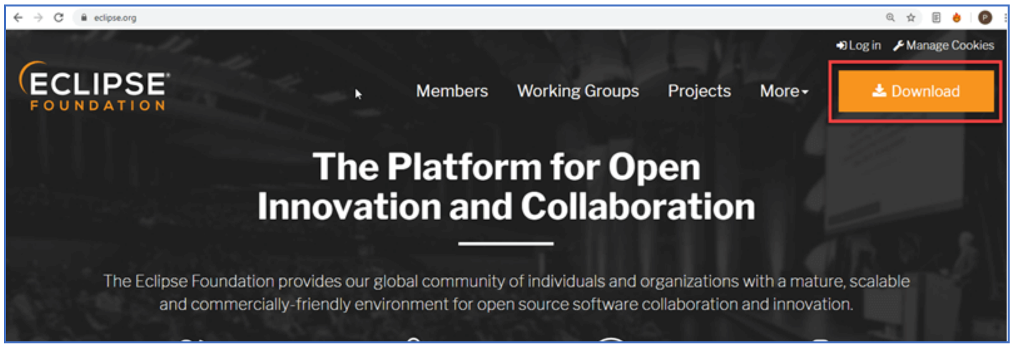
Following is a step by step guide to download and install Eclipse IDE:

Step 1) Installing Eclipse

Open your browser and type **https://www.eclipse.org/**

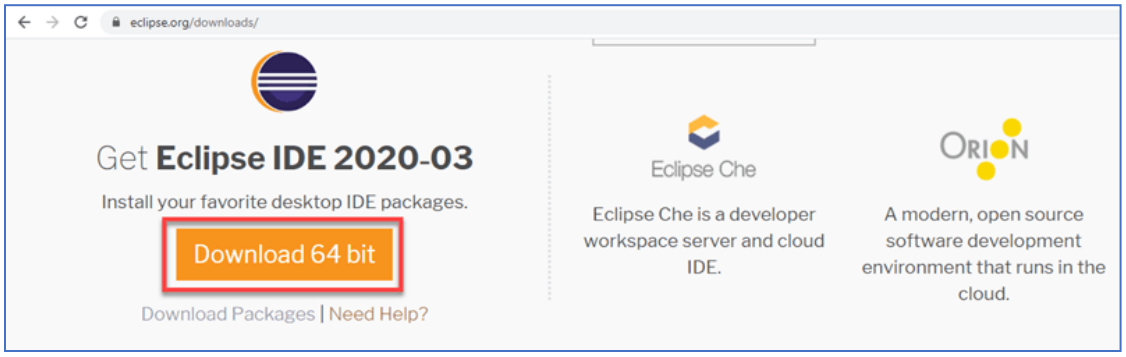


Step 2) Click on "**Download**" button.



The Platform for Open Innovation and Collaboration

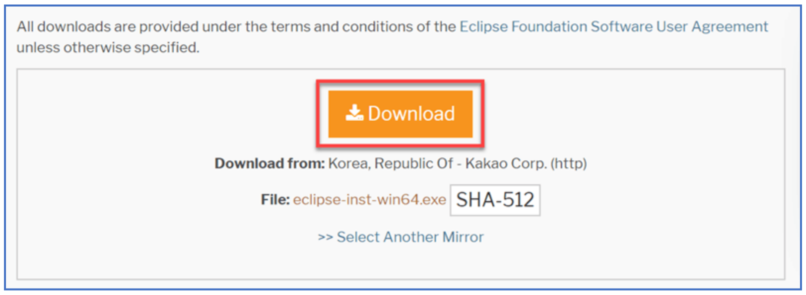
The Eclipse Foundation provides our global community of individuals and organizations with a mature, scalable and commercially-friendly environment for open source software collaboration and innovation.



Eclipse Che is a developer workspace server and cloud IDE.

A modern, open source software development environment that runs in the cloud.

Step 4) Click on "**Download**" button

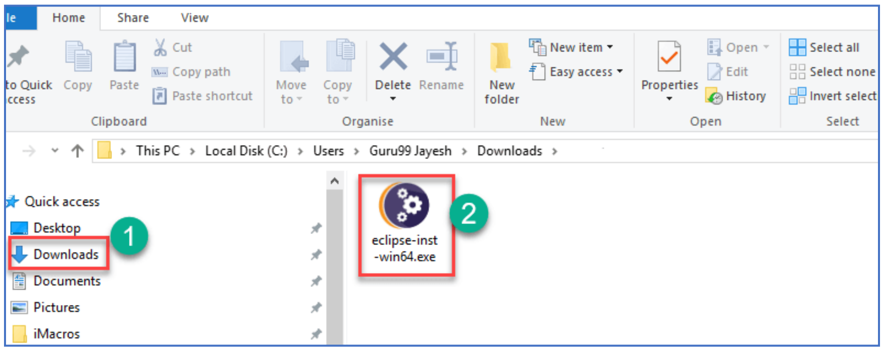

All downloads are provided under the terms and conditions of the Eclipse Foundation Software User Agreement unless otherwise specified.

Download from: Korea, Republic Of - Kakao Corp. (http)

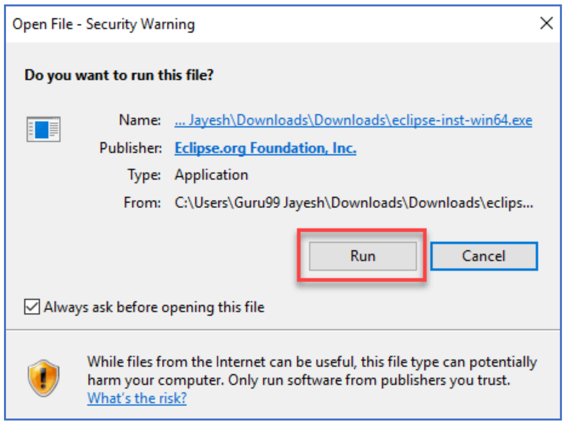
File: eclipse-inst-win64.exe SHA-512

Step 4) Install Eclipse.

1. Click on "**downloads**" in Windows file explorer.
2. Click on "**eclipse-inst-win64.exe**" file.



Step 5) Click on **Run** button


Open File - Security Warning

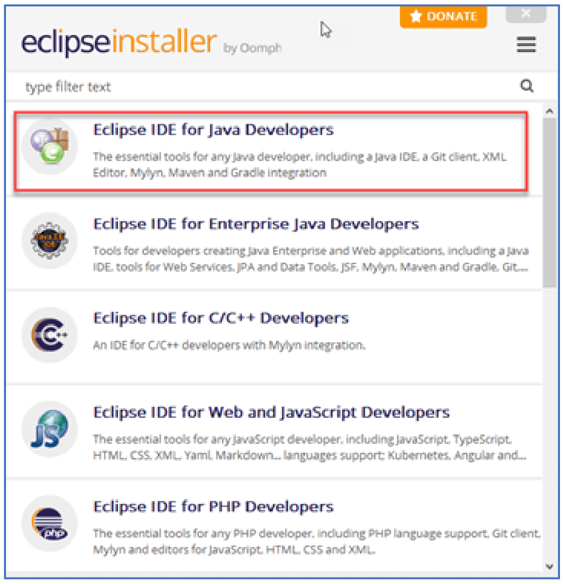
Do you want to run this file?

Name: ... Jayesh\Downloads\Downloads\eclipse-inst-win64.exe
Publisher: Eclipse.org Foundation, Inc.
Type: Application
From: C:\Users\Guru99 Jayesh\Downloads\Downloads\eclips...

☑ Always ask before opening this file

While files from the Internet can be useful, this file type can potentially harm your computer. Only run software from publishers you trust. What's the risk?

Step 6) Click on "**Eclipse IDE for Java Developers**"



Eclipse IDE for Java Developers
The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven and Gradle integration

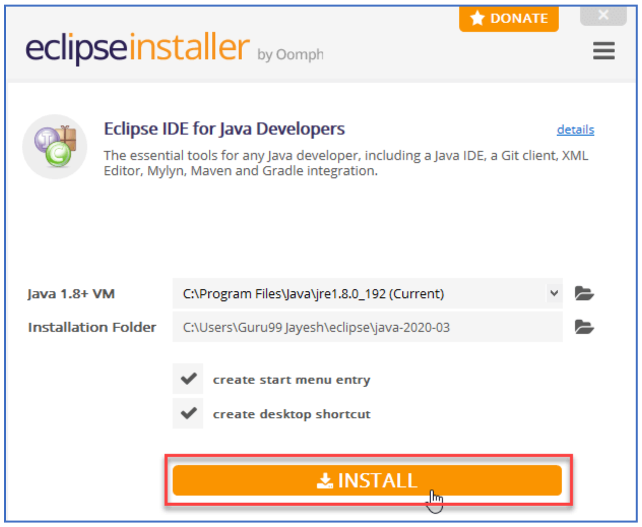
Eclipse IDE for Enterprise Java Developers
Tools for developers creating Java Enterprise and Web applications, including a Java IDE, tools for Web Services, JPA and Data Tools, JSF, Mylyn, Maven and Gradle, Git,...

Eclipse IDE for C/C++ Developers
An IDE for C/C++ developers with Mylyn integration.

Eclipse IDE for Web and JavaScript Developers
The essential tools for any JavaScript developer, including JavaScript, TypeScript, HTML, CSS, XML, Yaml, Markdown... languages support; Kubernetes, Angular and...

Eclipse IDE for PHP Developers
The essential tools for any PHP developer, including PHP language support, Git client, Mylyn and editors for JavaScript, HTML, CSS and XML.

Step 7) Click on "**INSTALL**" button

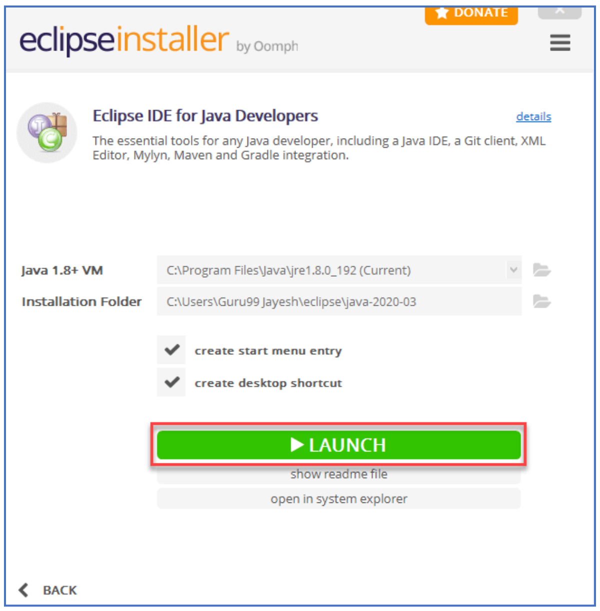


Eclipse IDE for Java Developers
The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven and Gradle integration.

Java 1.8+ VM: C:\Program Files\Java\jre1.8.0_192 (Current)
Installation Folder: C:\Users\Guru99 Jayesh\eclipse\java-2020-03

✓ create start menu entry
✓ create desktop shortcut

Step 8) Click on "**LAUNCH**" button.



The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven and Gradle integration.

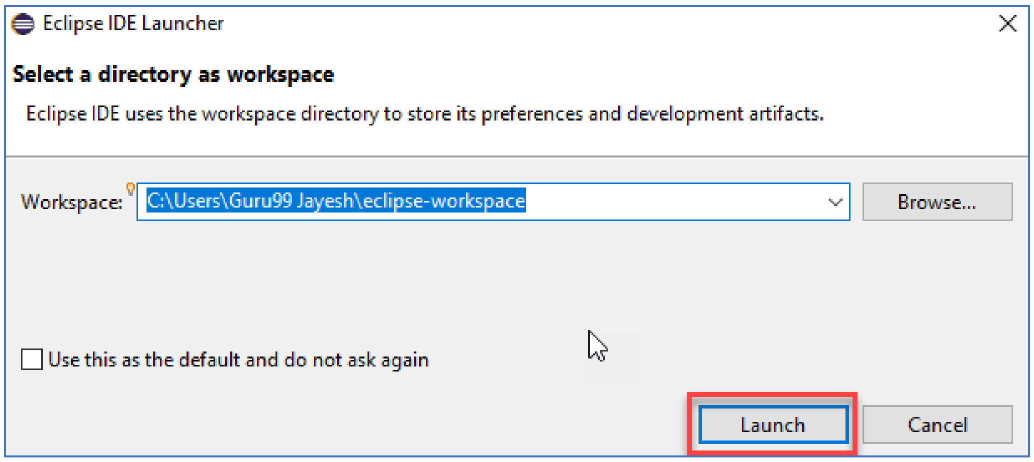
Java 1.8+ VM: C:\Program Files\Java\jre1.8.0_192 (Current)

Installation Folder: C:\Users\Guru99 Jayesh\eclipse\java-2020-03

✓ create start menu entry

✓ create desktop shortcut

Step 9) Click on "**Launch**" button.



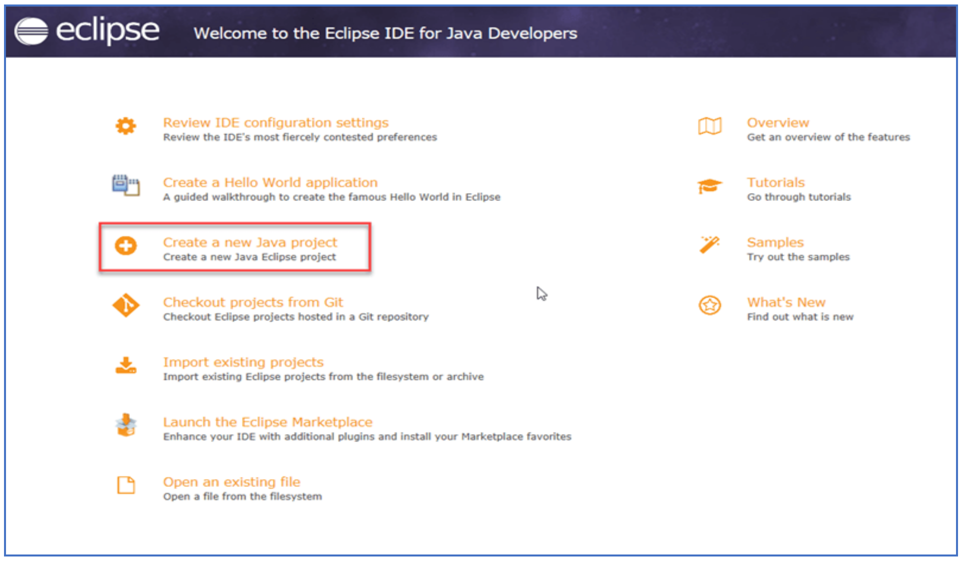
Select a directory as workspace

Eclipse IDE uses the workspace directory to store its preferences and development artifacts.

Workspace: C:\Users\Guru99 Jayesh\eclipse-workspace

☐ Use this as the default and do not ask again

Step 10) Click on “Create a new Java project” link.



eclipse Welcome to the Eclipse IDE for Java Developers

Review IDE configuration settings
Review the IDE's most fiercely contested preferences

Create a Hello World application
A guided walkthrough to create the famous Hello World in Eclipse

Create a new Java project
Create a new Java Eclipse project

Checkout projects from Git
Checkout Eclipse projects hosted in a Git repository

Import existing projects
Import existing Eclipse projects from the filesystem or archive

Launch the Eclipse Marketplace
Enhance your IDE with additional plugins and install your Marketplace favorites

Open an existing file
Open a file from the filesystem

Overview
Get an overview of the features

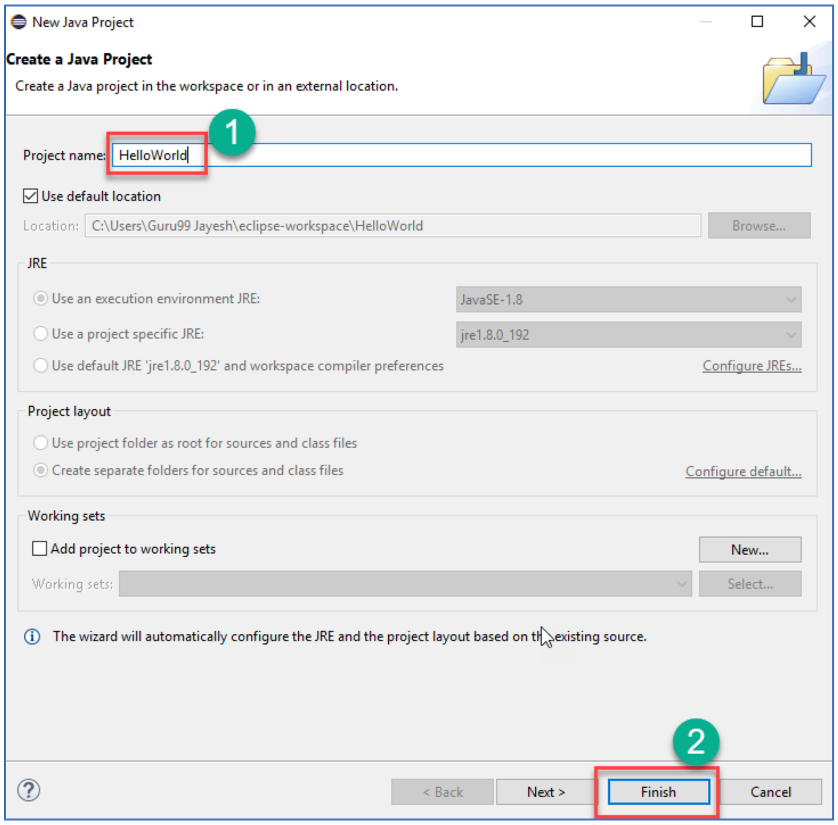
Tutorials
Go through tutorials

Samples
Try out the samples

What's New
Find out what is new

Step 11) Create a new Java Project

1. Write project name.
2. Click on “Finish button”.



New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: HelloWorld

☑ Use default location

Location: C:\Users\Guru99 Jayesh\eclipse-workspace\HelloWorld

JRE

● Use an execution environment JRE: JavaSE-1.8

○ Use a project specific JRE: jre1.8.0_192

○ Use default JRE 'jre1.8.0_192' and workspace compiler preferences

Project layout

○ Use project folder as root for sources and class files

● Create separate folders for sources and class files

Working sets

☐ Add project to working sets

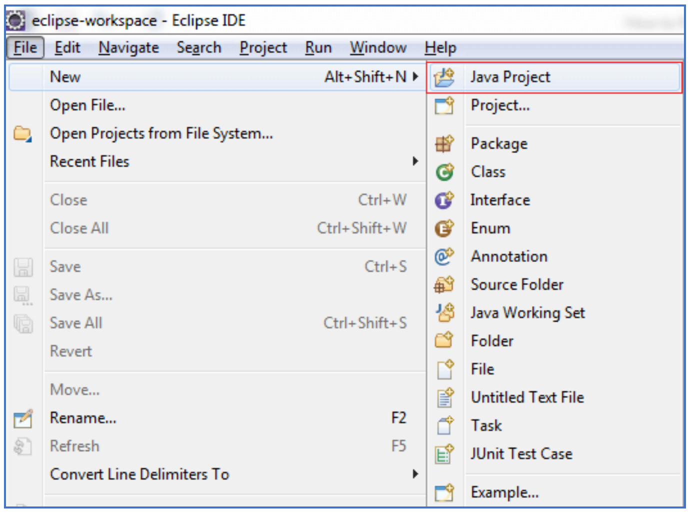
Working sets:

The wizard will automatically configure the JRE and the project layout based on the existing source.

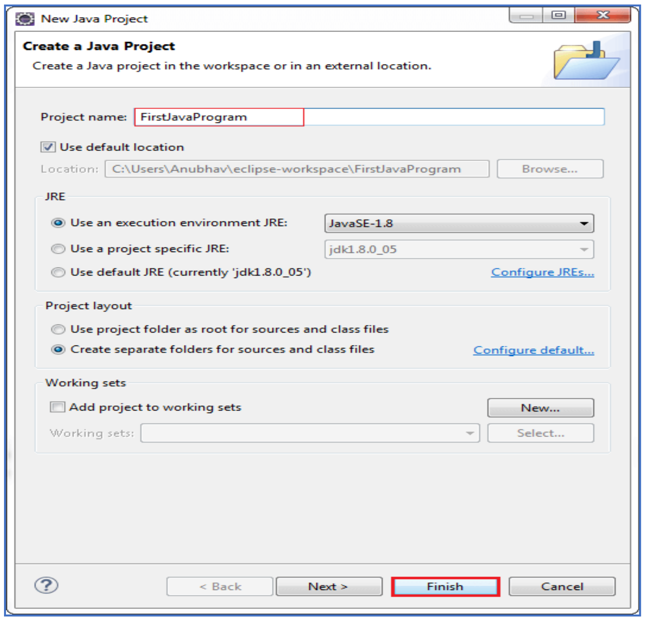
Exercise 03:
Execution of first java program using ECLIPSE

In this section, we learn how to run a Java program in eclipse step by step.

Step 1: Open Eclipse and click **File > New > Java Project**.



Step 2: Provide the **Project Name** and click on the **Finish** button.


New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: FirstJavaProgram

[✓] Use default location

Location: C:\Users\Anubhav\eclipse-workspace\FirstJavaProgram

JRE

(●) Use an execution environment JRE: JavaSE-1.8

( ) Use a project specific JRE: jdk1.8.0_05

( ) Use default JRE (currently 'jdk1.8.0_05')

Project layout

( ) Use project folder as root for sources and class files

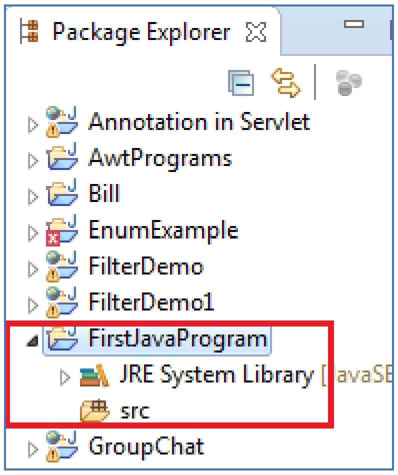
(●) Create separate folders for sources and class files

Working sets

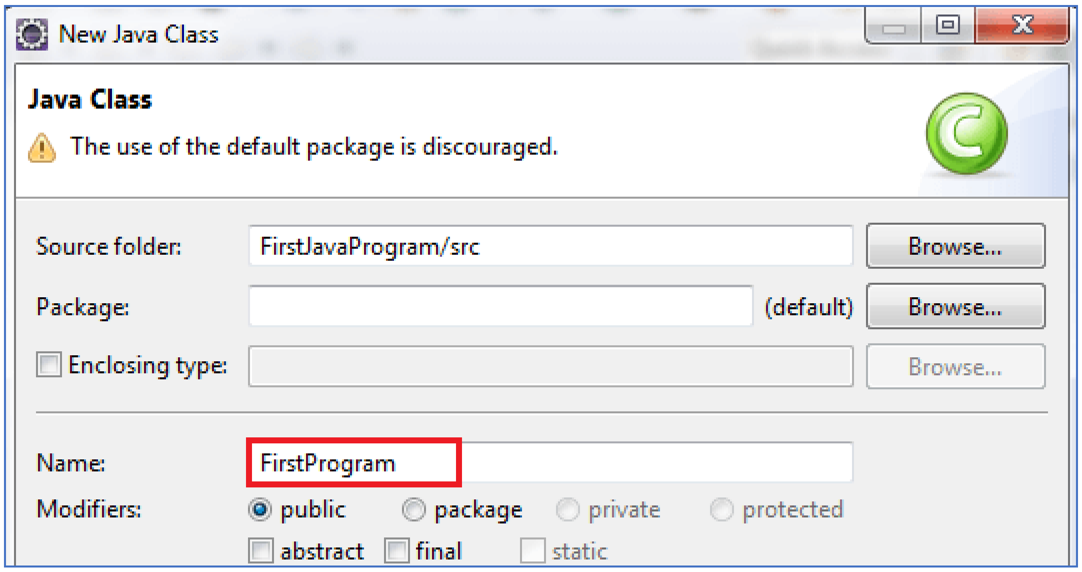
[ ] Add project to working sets

Working sets:

Step 3: In the **Package Explorer** (left-hand side of the window) select the project which you have created.



Step 4: Right-click on the **src** folder, select **New > Class** from the submenu. Provide the **Class name** and click on **Finish** button.


New Java Class

Java Class

The use of the default package is discouraged.

Source folder: FirstJavaProgram/src

Package: (default)

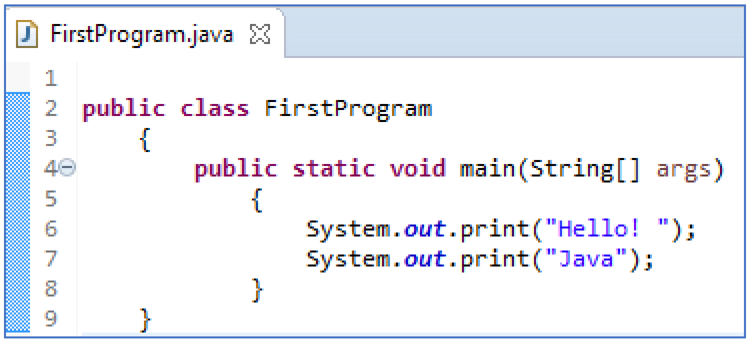
Enclosing type:

Name: FirstProgram

Modifiers: [✓ public] package private protected

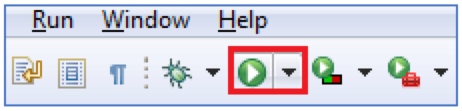
abstract final static

Step 5: Write the program and save it.

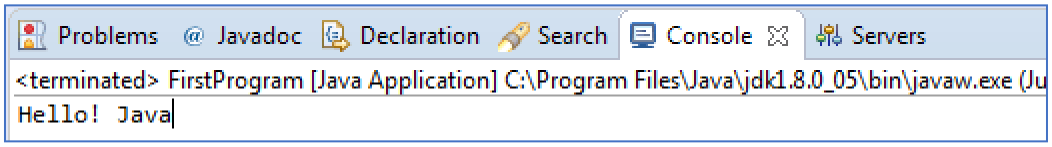


```java

public class FirstProgram
    {
        public static void main(String[] args)
            {
                System.out.print("Hello! ");
                System.out.print("Java");
            }
    }
```

Step 6: Now, press **Ctrl+F11** or click on the **Run** menu and select **Run** or click on Run button.



Step 7: Output



```
<terminated> FirstProgram [Java Application] C:\Program Files\Java\jdk1.8.0_05\bin\javaw.exe (Ju
Hello! Java
```

Execution of first java program using COMMAND LINE

Step 1:

Write a program on the notepad and save it with **.java** (for example, DemoFile.java) extension.

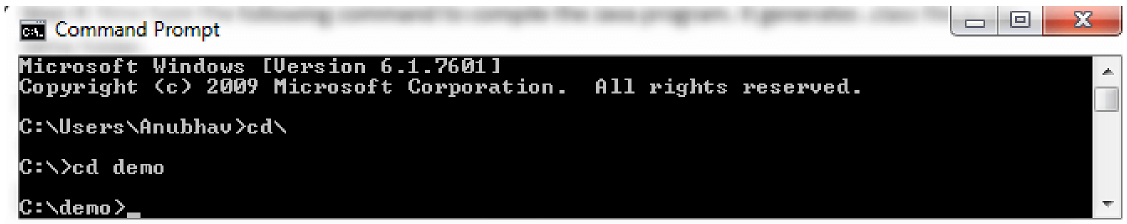
```
class DemoFile
{
public static void main(String args[])
        {
        System.out.println("Hello!");
        System.out.println("Java");
        }
}
```

Step 2:

Open Command Prompt.

Step 3:

Set the directory in which the .java file is saved. In our case, the .java file is saved in C:\demo.



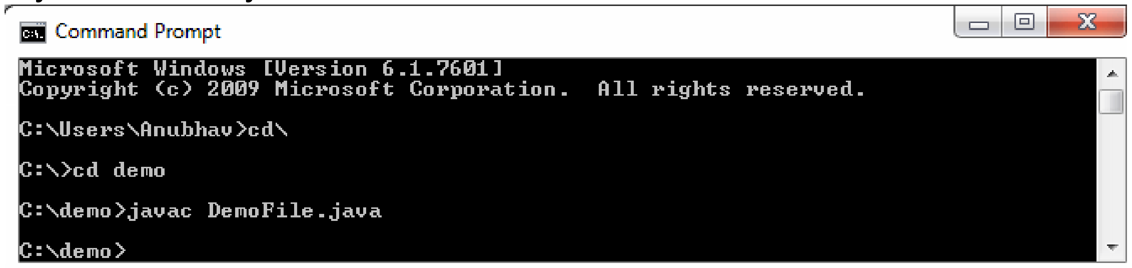
Command Prompt

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Anubhav>cd\

C:\>cd demo

C:\demo>_
```

Step 4:

Use the following command to compile the Java program. It generates a .class file in the same folder. It also shows an error if any.

javac DemoFile.java

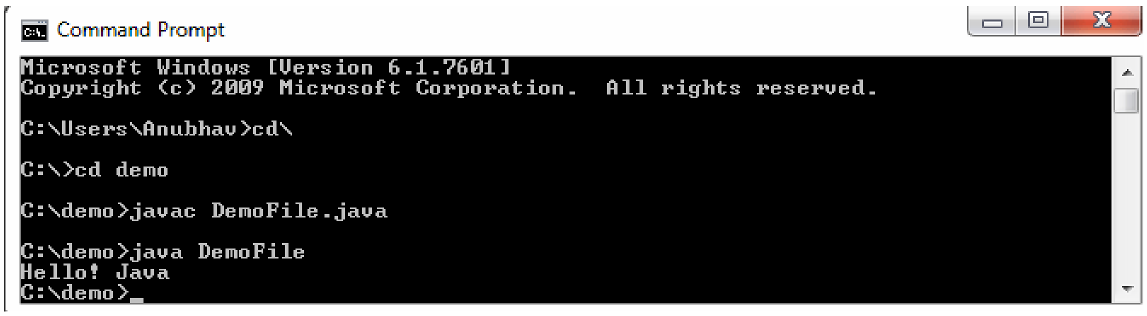


```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Anubhav>cd\

C:\>cd demo

C:\demo>javac DemoFile.java

C:\demo>
```

Step 5:

Use the following command to run the Java program:

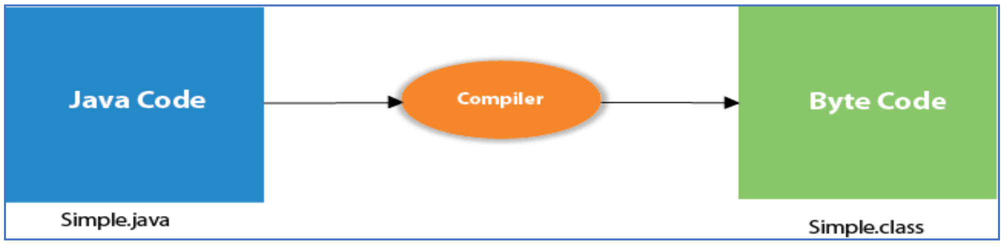
java DemoFile



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Anubhav>cd\

C:\>cd demo

C:\demo>javac DemoFile.java

C:\demo>java DemoFile
Hello! Java
C:\demo>_
```

Exercise 04:
Java code execution Process

At compile time, the Java file is compiled by Java Compiler (It does not interact with OS) and converts the Java code into bytecode.

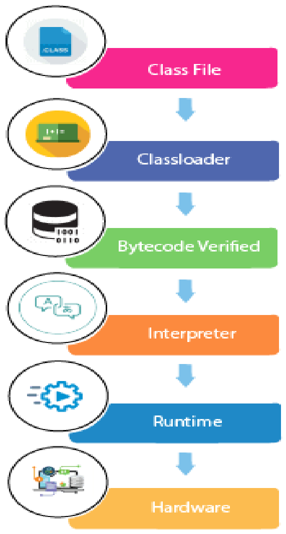


To compile: javacSimple.java

To execute: java Simple

What happens at runtime?

At runtime, the following steps are performed:



Classloader: It is the subsystem of JVM that is used to load class files.

Bytecode Verifier: Checks the code fragments for illegal code that can violate access rights to objects.

Interpreter: Read bytecode stream then execute the instructions

Program 1:

Write a java program to add two numbers.

```
class Add
{
   int no1;
   int no2;

   void get_numbers(int n1, int n2)
   {
      no1 = n1;
      no2 = n2;
         }

         intadd_numbers()
         {
            int sum;
            sum = no1 + no2;
            return(sum);
         }
}

class AddNos
{
   public static void main(String args[])
         {
                  int result = 0;
            Add AddObj = new Add();
            AddObj.get_numbers(25, 35);
            result = AddObj.add_numbers();
            System.out.println("The sum of two no's: " + result);
         }
}
```

Output:

```
The sum of two no's:60
```

Program 2:

Write a program to demonstrate the use of 'static' variable.

Program A

```
class Student
{
         int rollno;//instance variable
         String name;
         static String college_Name;//static variable
```

```
        void get_Student_Details(int r, String n)
        {
                rollno = r;
                name = n;
        }
        //method to display the values
        void display ()
        {
                System.out.println(rollno+" "+name+" "+ college_Name);
        }
}
//Test class to show the values of objects
public class TestStaticVariable
{
        public static void main(String args[])
        {
                Student s1 = new Student();
                Student s2 = new Student();
                Student s3 = new Student();
                s1. get_Student_Details (111,"Karan");
                s2. get_Student_Details (222,"Aryan");
                s3. get_Student_Details (333,"Simran");
                //we can change the college of all objects by the single line of code
                Student.college="BBDIT";
                s1.display();
                s2.display();
                s3.display();
        }
}
```

Output

```
111 Karan BBDIT

222 Aryan BBDIT

333 Simran BBDIT
```

Program B

```
class VariableDemo
{
        static int count=0;
        public void increment()
        {
                count++;
        }
        public static void main(String args[])
        {
                VariableDemo obj1=new VariableDemo();
                VariableDemo obj2=new VariableDemo();
                obj1.increment();
                obj2.increment();
                System.out.println("Obj1: count is=" + obj1.count);
                System.out.println("Obj2: count is=" + obj2.count);
        }
}
```

Output

```
5

6

7
```

Program 3:

Identify and resolve issues in the given code snippet

```
class Calculate
{
        void cube(int x)
        {
          int result
                result = x*x*x;
        }
        void display()
        {
          System.out.println("Cube of a no: " + result)
        }



        public static void main(String args[])
        {
                cube(5);
                display();
        }
}
```

Output

```
Error: cannot file symbol
```

```
System.out.println("Cube of  no:"+result);
```

Program 4:
Write a program to demonstrate the use of default constructor.

```
        //Java Program to create and call a default constructor
        class Bike
        {
                //creating a default constructor
                Bike()
                {
                        System.out.println("Bike is created");
                }
                //main method
                public static void main(String args[])
                {
                        //calling a default constructor
                        Bike b=new Bike();
                }
}
```

Output:

```
Bike is created
```

Program 5:

Write a program to demonstrate the use of parameterized constructor.

```
class Student
{
        int id;
        String name;
        //creating a parameterized constructor
        Student(int i, String n)
        {
                id = i;
                name = n;
        }
        //method to display the values
        void display()
        {
                System.out.println(id+" "+name);
        }
        public static void main(String args[])
        {
                //creating objects and passing values
                Student s1 = new Student(111,"Karan");
                Student s2 = new Student(222,"Aryan");
                //calling method to display the values of object
                s1.display();
                s2.display();
        }
}
```

Output:

```
111  Karan
222  Aryan
```

Program 6:

Find the error in the program

```
public class A
{
        private int data=40;
        private void msg()
        {
                System.out.println("Hello java");
        }
}

public class Simple
{
        public static void main(String args[])
        {
                A obj=new A();
                System.out.println(obj.data);
                obj.msg();
        }
}
```

Error:

Program 7:

Write a program to demonstrate the use of 'this' keyword.

Program A: we are using **'this'** keyword to distinguish local variable and instance variable

```
class Student
{
        int rollno;
        String name;
        float fee;
        Student(int rollno, String name, float fee)
        {
                this.rollno=rollno;
                this.name=name;
                this.course=course;
        }
        void display()
        {
                System.out.println(rollno+" "+name+" "+fee);
        }
}

class ThisMain
{
        public static void main(String args[])
        {
                Student s1=new Student(111,"ankit",5000f);
                Student s2=new Student(112,"sumit",6000f);
                s1.display();
                s2.display();
        }
}
```

Output:

```
111
ankit
5000
 112
sumit
6000
```

Program B: **this()** : to invoke current class constructor

```
class Student
{
		int rollno;
		String name, course;
		float fee;
		Student(int rollno, String name, String course)
		{
			this.rollno=rollno;
			this.name=name;
			this.course=course;
		}
		Student(int rollno, String name, String course, float fee)
		{
			this(rollno, name, course);
			this.fee=fee;
		}
		void display()
		{
			System.out.println(rollno+" "+name+" "+course+" "+fee);
		}
}
class ThisMain
{
		public static void main(String args[])
		{
			Student s1=new Student(111,"ankit","java");
			Student s2=new Student(112,"sumit","java",6000f);
			s1.display();
			s2.display();
		}
}
```

Output:

```
111  ankit   java
112  sumit   java  6000
```

Program 8:
Write a Java program to convert primitive data type into objects (Autoboxing)

```
public class AutoboxingDemo
{
	public static void main(String args[])
	{
		//Converting int into Integer
		int a=20;
		Integer j=a; //autoboxing
```

```
                System.out.println(a+"   "+"   "+j);
        }
}
```

Output:

20   20

Program 9:

Write a Java program to convert object into primitive data type (Unboxing)

```
public class UnboxingDemo
{
        public static void main(String args[])
        {
                //Converting Integer to int
                Integer a=new Integer(3);
                int j=a; //unboxing
                System.out.println(a+"   "+"   "+j);
        }
}
```

Output:

```
3  3
```

Program 10:

Write the output of the following programs

(i)

```
class op
{
   public static void main(String args[])
   {
      int a = 10, inti = 3, int j = 15;
      System.out.println(-3-a+j);
   }
}
```

Output:

```
2
```

(ii)

```
class Precedence
{
            public static void main(String[] args)
            {
                  int a = 10, b = 5, c = 1, result;
                  result = a-++c-++b;
                  System.out.println(result);
            }
}
```

Output:

```
2
```

Program 11:

Write a java program to implement read-only class.

```
class Employee
{
        private String name = “Ramesh”;
        public String getName()
        {
                return name;
        }
}

public class ReadOnly
{
        public static void main(String[] args)
        {
                Employee e = new Employee();
                System.out.println(e.getName());
        }
}
```

Output:

```
Ramesh
```

Program 12:

Write a java program to implement read-write class

```
class Employee
{
        private String name;
        public String getName()
        {
                return name;
        }
        public void setName(String name)
        {
                this.name=name ;
        }
}
public class ReadWrite
{
        public static void main(String[] args)
        {
                Employee e = new Employee();
                e.setName("Charles");
                System.out.println(e.getName());
        }
}
```

Output:

```
Charles
```

Program 13:

Write a program to find minimum no in a list of numbers.

```
class Minimum
{
        //creating a method which receives an array as a parameter
        void min(int arr[])
        {
                int min=arr[0];
                for(int i=1;i <arr.length;i++)
                {
                        if(min>arr[i])
                           min=arr[i];
                }
                System.out.println(min);
        }
}
class MinMain
{
        public static void main(String args[])
        {
                int a[]={40, 51, 25, 15, 100, 50, 10,212};
                Minimum M = new Minimum();
                M.min(a);//passing array to method
        }
}
```

Output:

```
10
```

_Program 14:

Write a Java Program to return an array from the method

```
class ReturnArray

{

//creating method which returns an array
public int[] get()

{

inta[] = {10,30,50,90,60};

return (a);

}

}

class ArrayReturn

{

public static void main(String args[])

{

ReturnArray R = new ReturnArray();

//calling method which returns an array

Int  arr[] =R.get();

//printing the values of an array

for(inti=0; i<arr.length;i++)

System.out.println(arr[i]);

}

}
```

Output:

Program 15:

Write a java program to find the largest of 3 numbers.

```
class Largest
{
        public static void main(String args[])
        {
                int a = 10, b = 20, c = 30;
                if ( a > b && a > c)
                {
                        System.out.println ("a is greater");
                }
                else if ( b > a && b > c )
                {
                        System.out.println ("b is greater");
                }
                else
                {
                        System.out.println ("c is greater");
                }
        }
}
```

Output:

C is greater

Program 16:

Write a java program of grading system for fail, Second class, First class & FCD.

```
public class GradingSystem
{
    public static void main(String[] args)
    {
            int marks=65;
            if(marks<35)
            {
                    System.out.println("fail");
            }
            else if(marks>=35 && marks<60)
            {
                    System.out.println("2nd Class");
            }
            else if(marks>=60 && marks<85)
            {
                    System.out.println("First Class");
            }
            else if(marks>=85 && marks<=100)
            {
                    System.out.println("FCD");
            }
            else
            {
                    System.out.println("Invalid marks!");
            }
    }
}
```

Output:

```
First Class
```

Program 17:

Write a java program to display the day for the given day number.

```
public class Day
{
	public static void main(String[] args)
	{
		int day = 5;
		String dayString;

		switch (day)
		{
			case 1:
				dayString = "Monday";
				break;

			case 2:
				dayString = "Tuesday";
				break;

			case 3:
				dayString = "Wednesday";
				break;

			case 4:
				dayString = "Thursday";
				break;

			case 5:
				dayString = "Friday";
				break;

			case 6:
				dayString = "Saturday";
				break;

			case 7:
				dayString = "Sunday";
				break;

			default:
				dayString = "Invalid day";
		}
		System.out.println(dayString);
	}
}
```

Output: Friday

Program 18:

Write a java program to calculate the sum of odd and even numbers till 100

```
class Sum_Odd_Even
{
        public static void main(String[] args)
        {
                int  sumE = 0, sumO = 0;
                for(inti = 0; i<= 100; i++)
                {
                        if(i % 2 == 0)
                        {
                                sumE = sumE + i;
                        }
                        else
                        {
                                sumO = sumO + i;
                        }
                }
                System.out.println("Sum of Even Numbers:" + sumE);
                System.out.println("Sum of Odd Numbers:" + sumO);
        }
}
```

Output:

```
Sum of Even Numbers:2550
Sum of Odd Numbers:2500
```

Program 19:

write a java program check whether the given String is Palindrome or not.

```
import java.util.Scanner;

class PalindromeTest {
   public static void main(String args[])
   {
      String reverseString="";
      Scanner scanner = new Scanner(System.in);

      System.out.println("Enter a string to check if it is a palindrome:");
      String inputString = scanner.nextLine();

      int length = inputString.length();

      for ( int i = length - 1 ; i >= 0 ; i-- )
         reverseString = reverseString + inputString.charAt(i);

      if (inputString.equals(reverseString))
         System.out.println("Input string is a palindrome.");
      else
         System.out.println("Input string is not a palindrome.");

   }
}
```

Output 1:

```
Enter any string:abccba
The input String is a palindrome.
```

Output 2:

```
Enter any string:abcdef
The input String is not a palindrome.
```

Program 20:

Java Program to illustrate Open Closed Principle

```
class Cuboid
 {
    public double length;
    public double breadth;
    public double height;
```

```
}

class Application
 {
 public double get_total_volume(Cuboid[] geo_objects)
  {
   double vol_sum = 0;
 for (Cuboid geo_obj : geo_objects)
{
vol_sum += geo_obj.length * geo_obj.breadth

                       * geo_obj.height;

       }
   return vol_sum;
}
}

public class GFG
 {
 public static void main(String args[])
  {
    Cuboid cb1 = new Cuboid();

      cb1.length = 5;
      cb1.breadth = 10;
      cb1.height = 15;
   Cuboid cb2 = new Cuboid();
       cb2.length = 2;
       cb2.breadth = 4;
       cb2.height = 6;
    Cuboid cb3 = new Cuboid();
       cb3.length = 3;
       cb3.breadth = 12;
       cb3.height = 15;
        Cuboid[] c_arr = new Cuboid[3];

        c_arr[0] = cb1;

        c_arr[1] = cb2;

        c_arr[2] = cb3;
    Application app = new Application();
    double vol = app.get_total_volume(c_arr);
    System.out.println("The total volume is " + vol);
  }
}
Output:
The total volume is 1338.0
```

Program 21:

Program to illustrate of method overloading

```
class Helper {

    static int Multiply(int a, int b)

    {

        return a * b;

   }

    static int Multiply(int a, int b, int c)

    {

        return a * b * c;

    }
}

class GFG {

    public static void main(String[] args)

    {
        System.out.println(Helper.Multiply(2, 4));

        System.out.println(Helper.Multiply(2, 7, 3));

    }
}

OUTPUT:
 8
42
```

Program 22:

Program to illustrate method overriding

```
class Animal {
  public void displayInfo() {
    System.out.println("I am an animal.");
  }
}

class Dog extends Animal {
  @Override
  public void displayInfo() {
    System.out.println("I am a dog.");
  }
}

class Main {
  public static void main(String[] args) {
    Dog d1 = new Dog();
    d1.displayInfo();
  }
}

super.displayInfo();
```

Output :

```
I am a dog
I am an Animal
```

Program 23:

Java program to demonstrate application of employee salary.

```
 class Employee {

    public static int base = 10000;

    int salary()

    {

        return base;
    }
}


class Manager extends Employee {

    int salary()

    {

        return base + 20000;

    }
}



class Clerk extends Employee {

    // This method overrides salary() of Parent

    int salary()

    {

        return base + 10000;

    }
}


// Driver class

class Main {

    static void printSalary(Employee e)

    {
     System.out.println(e.salary());

    }
    public static void main(String[] args)
    {

        Employee obj1 = new Manager() ;

        System.out.print("Manager's salary : ");

        printSalary(obj1);

          Employee obj2 = new Clerk();

        System.out.print("Clerk's salary : ");

        printSalary(obj2);

    }
}
```

```
Output:
Manager's salary : 30000
Clerk's salary : 20000
```

Program 24:

Java program to demonstrate simple calculator.

```
interface Add_Sub {

    public void add(double x, double y);

    public void subtract(double x, double y);
}

interface Mul_Div {

    public void multiply(double x, double y);

    public void divide(double x, double y);
}


interface Calculator extends Add_Sub, Mul_Div {

    public void printResult(double result);
}
```

```
public class MyCalculator implements Calculator {
   public void add(double x, double y)

   {

      double result = x + y;

      printResult(result);

  }
   public void subtract(double x, double y)

   {

      double result = x - y;

      printResult(result);

   }


   public void multiply(double x, double y)

   {

      double result = x * y;

      printResult(result);

   }

   public void divide(double x, double y)

   {

      double result = x / y;

      printResult(result);

   }

   public void printResult(double result)
```

```
    {

        System.out.println(

            "The result is : " + result);

    }

    public static void main(String args[])

    {

        MyCalculator c = new MyCalculator();

        c.add(5, 10);

        c.subtract(35, 15);

        c.multiply(6, 9);

        c.divide(45, 6);

    }
}
```

Output:

```
The result is : 15.0
The result is : 20.0
The result is : 54.0
The result is : 7.5
```

Program 25:

program to illustrate get file information

```
import java.io.File;

public class GetFileInfo {
  public static void main(String[] args) {
    File myObj = new File("filename.txt");
    if (myObj.exists()) {
      System.out.println("File name: " + myObj.getName());
      System.out.println("Absolute path: " + myObj.getAbsolutePath());
      System.out.println("Writeable: " + myObj.canWrite());
      System.out.println("Readable " + myObj.canRead());
      System.out.println("File size in bytes " + myObj.length());
    } else {
      System.out.println("The file does not exist.");
    }
  }
}
```

```
Output:
File name: filename.txt
Absolute path: C:/User/admin/Document/filename.txt
Writeable: true
Readable true
File size in bytes 17
```

Program 26:

program to illustrate an Exception handling in java

```
class Main
{
   public static void main(String args[]){

     try{

         try{
            System.out.println("Try Block1");
            int num =15/0;
            System.out.println(num);
         }
         catch(ArithmeticException e1){
            System.out.println("Block1 Exception: e1");
         }

         try{
            System.out.println("Try Block2");
            int num =100/0;
            System.out.println(num);
         }
         catch(ArrayIndexOutOfBoundsException e2){
            System.out.println("Block2 Exception: e2");
         }
        System.out.println("General statement after Block1 and Block2");
    }
    catch(ArithmeticException e3){
      System.out.println("Main Block Arithmetic Exception");
    }
    catch(ArrayIndexOutOfBoundsException e4){
     System.out.println("Main Block ArrayIndexOutOfBoundsException");
    }
    catch(Exception e5){
     System.out.println("Main Block General Exception");
    }
finally {
            System.out.println (":: Finally Block::");
            System.out.println ("No Exception::finally block executed");
        }

    System.out.println("Code after Nested Try Block");
   }
```

```
Output:
Try Block1
Block1 Exception:e1
Try Block2
Main Block Arithmetic Exception
```

Program 27:

Java program to illustration of Interface segregation principal

```
 interface Toy {
    void setPrice(double price);
    void setColor(String color);
}

interface Movable {
   void move();
}

interface Flyable {
   void fly();
}
class ToyHouse implements Toy {
   double price;
   String color;
   public void setPrice(double price) {
      this.price = price;
   }
   public void setColor(String color) {
      this.color=color;
   }
   public String toString(){
      return "ToyHouse: Toy house- Price: "+price+" Color: "+color;
   }
}
class ToyCar implements Toy, Movable {
   double price;
   String color;
   public void setPrice(double price) {
      this.price = price;
   }
   public void setColor(String color) {
    this.color=color;
   }
      public void move(){
      System.out.println("ToyCar: Start moving car.");
   }
      public String toString(){
      return "ToyCar: Moveable Toy car- Price: "+price+" Color: "+color;
   }
}
class ToyPlane implements Toy, Movable, Flyable {
   double price;
   String color;
   public void setPrice(double price) {
      this.price = price;
   }
   public void setColor(String color) {
      this.color=color;
   }
      public void move(){
      System.out.println("ToyPlane: Start moving plane.");
   }
     public void fly(){
      System.out.println("ToyPlane: Start flying plane.");
   }
   public String toString(){
      return "ToyPlane: Moveable and flyable toy plane- Price: "+price+" Color: "+color;
   }
}
public class ToyBuilder {
   public static void main (Strinf args{})
{
      ToyHouse toyHouse=new ToyHouse();
      toyHouse.setPrice(15.00);
      toyHouse.setColor("green");
      ToyCar toyCar=new ToyCar();
      toyCar.setPrice(25.00);
      toyCar.setColor("red");
      toyCar.move();
     ToyPlane toyPlane=new ToyPlane();
      toyPlane.setPrice(125.00);
      toyPlane.setColor("white");
      toyPlane.move();
      toyPlane.fly();
          }
}
```

Output;

```
ToyCar: start moving car
Toyplane: start moving plane
Toyplane: start flying plane
```