**Practical file submitted in partial fulfillment for the evaluation of**

**“Object Oriented Programming Lab   
(AIDS-252)”**



**Submitted By:**

**Student Name:** Rohit Kumar Saxena

**Enrolment no:** 03817711922

**Branch & Section:** AI-DS (A)

**Submitted To:**

* Ms. Shikha Jain

**Index**

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Program Title | Date | Sign |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Index**

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Program Title | Date | Sign |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Index**

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Program Title | Date | Sign |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

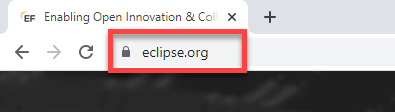
**Program 1**

**Program 1:** Getting Familiar with Eclipse:  
 (a) Download and Install Eclipse.  
 (b) Using Eclipse for Java.

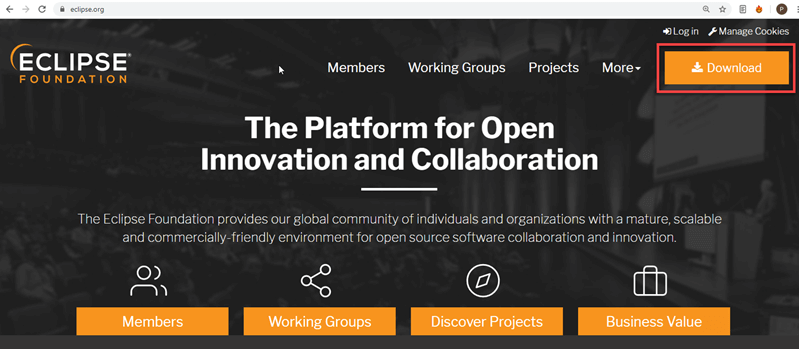
**Eclipse Download and Installation Steps:**

Step 1)**Installing Eclipse**

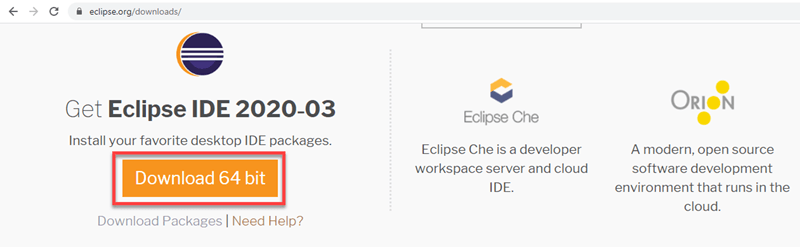
Open your browser and type <https://www.eclipse.org/>

[](https://www.guru99.com/images/2/041720_1109_HowtoDownlo1.png)

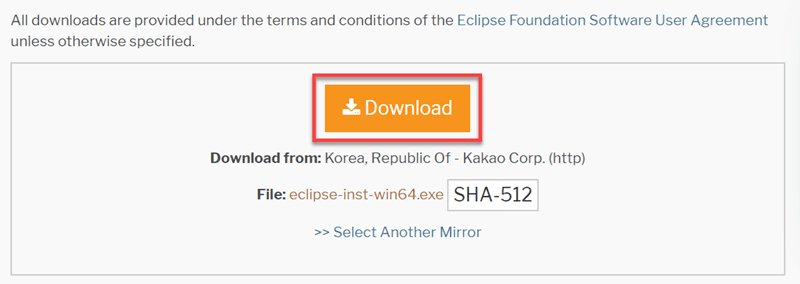
Step 2)**Click on “Download” button.**

[](https://www.guru99.com/images/2/041720_1109_HowtoDownlo2.png)

Step 3)**Click on “Download 64 bit” button**

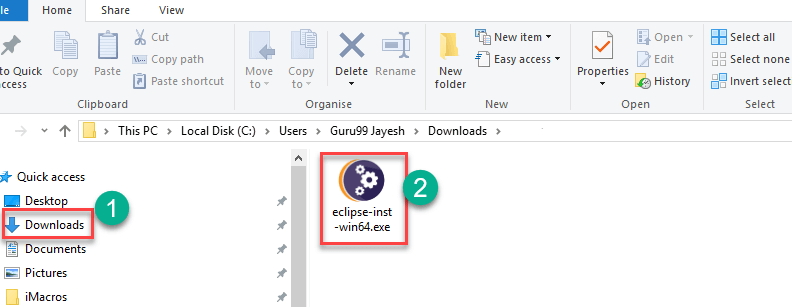
[](https://www.guru99.com/images/2/041720_1109_HowtoDownlo3.png)

Step 4)**Click on “Download” button**

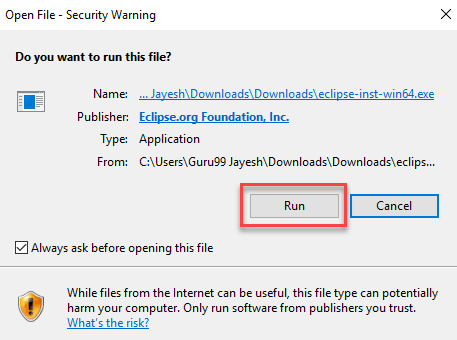
[](https://www.guru99.com/images/2/041720_1109_HowtoDownlo4.png)

Step 4) **Install Eclipse.**

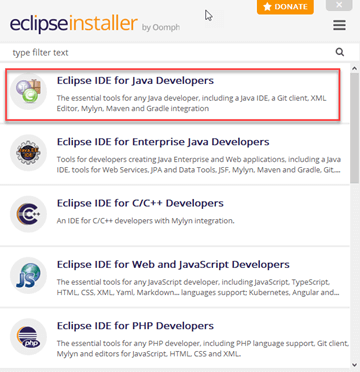
1. Click on “downloads” in Windows file explorer.
2. Click on “eclipse-inst-win64.exe” file.

[](https://www.guru99.com/images/2/041720_1109_HowtoDownlo5.png)

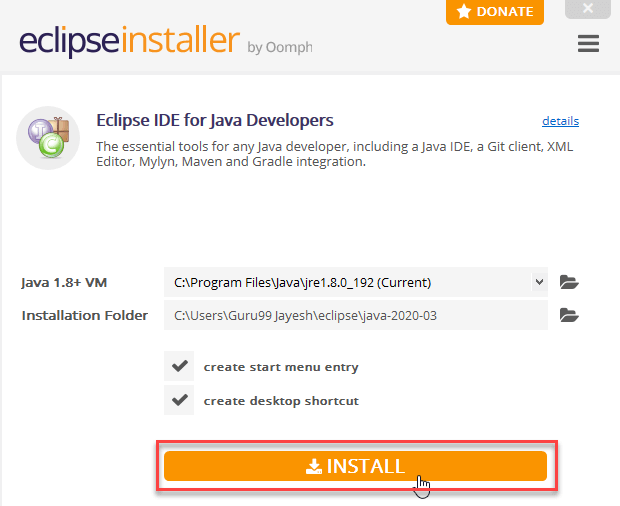
Step 5)**Click on Run button**

[](https://www.guru99.com/images/2/041720_1109_HowtoDownlo6.png)

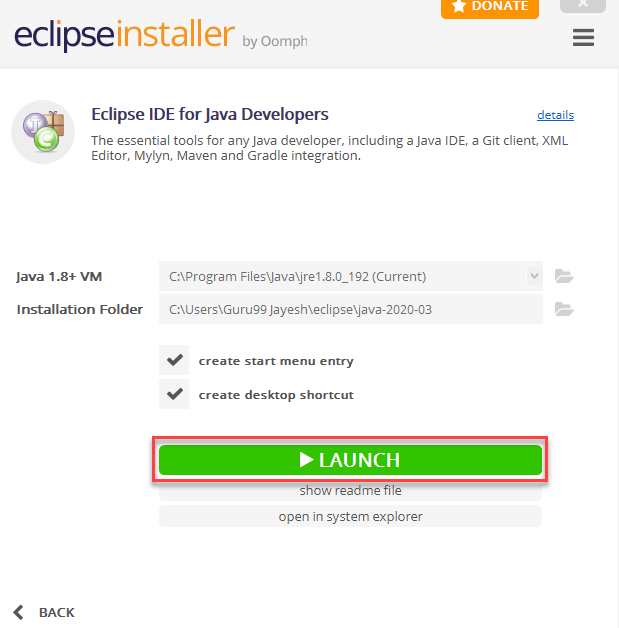
Step 6)**Click on “Eclipse IDE for Java Developers”**

[](https://www.guru99.com/images/2/041720_1109_HowtoDownlo7.png)

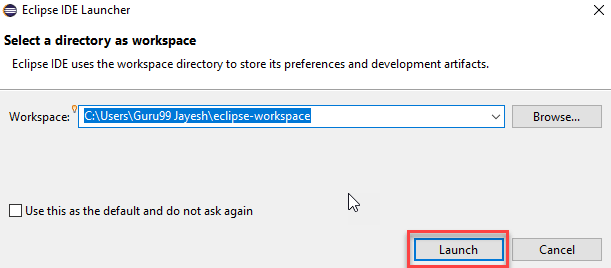
Step 7)**Click on “INSTALL” button**

[](https://www.guru99.com/images/2/041720_1109_HowtoDownlo8.png)

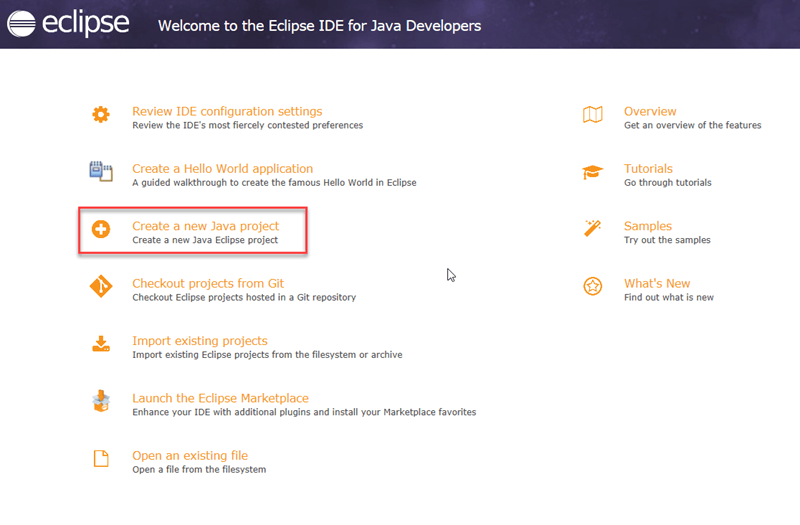
Step 8)**Click on “LAUNCH” button.**

[](https://www.guru99.com/images/2/041720_1109_HowtoDownlo9.png)

Step 9)**Click on “Launch” button.**

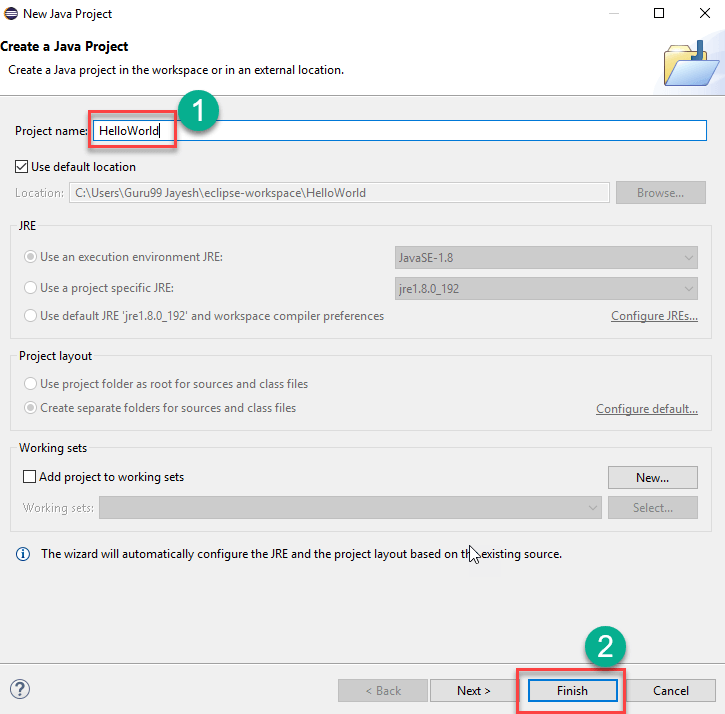
[](https://www.guru99.com/images/2/041720_1109_HowtoDownlo10.png)

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

[](https://www.guru99.com/images/2/041720_1109_HowtoDownlo11.png)

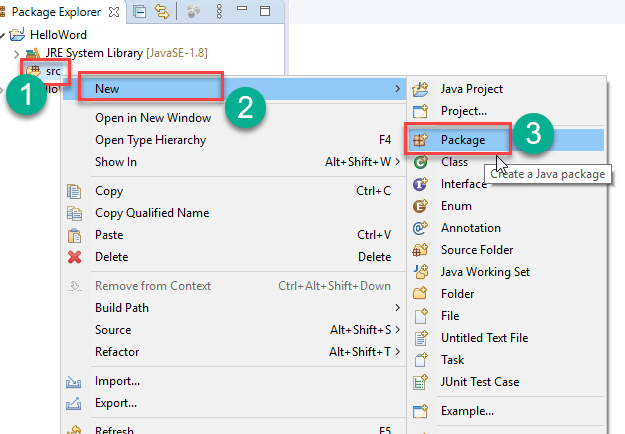
Step 11)**Create a new Java Project**

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

[](https://www.guru99.com/images/2/041720_1109_HowtoDownlo12.png)

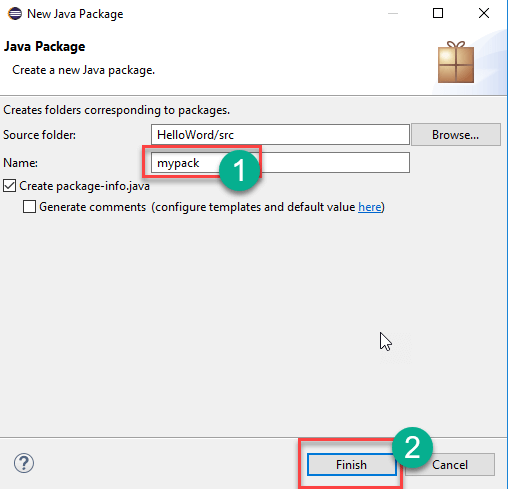
Step 12)[**Create Java Package**](https://www.guru99.com/java-packages.html)**.**

1. Goto “src”.
2. Click on “New”.
3. Click on “Package”.

[](https://www.guru99.com/images/2/041720_1109_HowtoDownlo13.png)

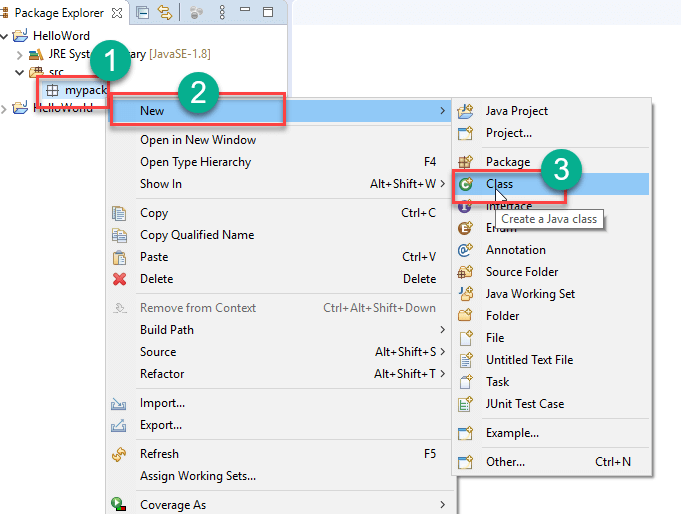
Step 13)**Writing package name.**

1. Write name of the package
2. Click on Finish button.

[](https://www.guru99.com/images/2/041720_1109_HowtoDownlo14.png)

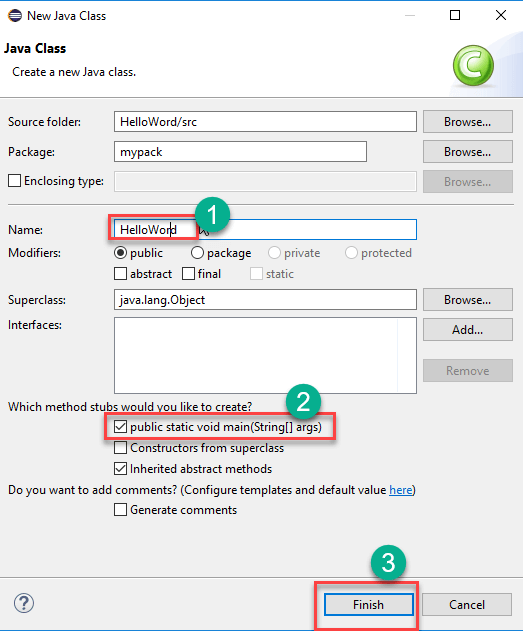
Step 14) **Creating Java Class**

1. Click on package you have created.
2. Click on “New”.
3. Click on “Class”.

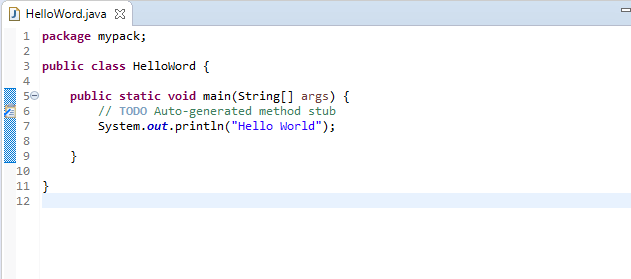
[](https://www.guru99.com/images/2/041720_1109_HowtoDownlo15.png)

Step 15)**Defining Java Class.**

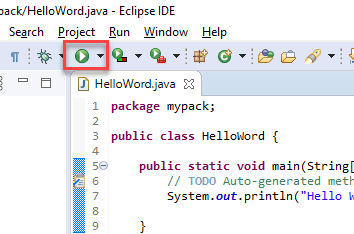
1. Write class name
2. Click on “public static void main (String[] args)” checkbox.
3. Click on “Finish” button.

[](https://www.guru99.com/images/2/041720_1109_HowtoDownlo16.png)

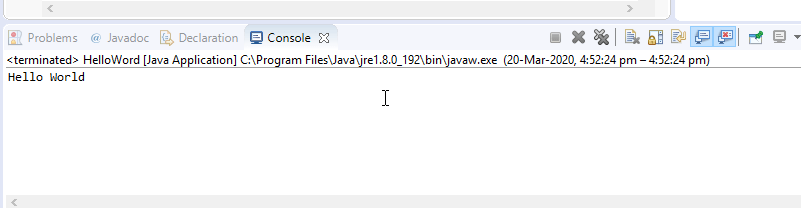
Helloword.java file will be created as shown below:

[](https://www.guru99.com/images/2/041720_1109_HowtoDownlo17.png)

Step 16)**Click on “Run” button.**

[](https://www.guru99.com/images/2/041720_1109_HowtoDownlo18.png)

Output will be displayed as shown below.

[](https://www.guru99.com/images/2/041720_1109_HowtoDownlo19.png)

**Learning outcome of the Program:**

|  |
| --- |
|  |
|  |
|  |
|  |

**Program 2**

**Program 2:** Write a Java program to print “Hello World” to understand compilation and execution of java program.

**Theory:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Code:**

public class helloWorld {

    public static void main(String[] args) {

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

    }

}

**Output:**

****

**Learning outcome of the Program:**

|  |
| --- |
|  |
|  |
|  |
|  |

**Program 3**

**Program 3:** Write java program demonstrating the usage of literal datatypes.

**Theory:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Code:**

public class literals {

    public static void main(String[] args) {

        int count = 987;

        float floatVal = 4534.99f;

        double cost = 19765.567;

        int hexaVal = 0x7e4;

        int binary = 0b11010;

        char alpha = 'p';

        String str = "Java";

        boolean boolVal = true;

        int octalVal = 067;

        String stuName = null;

        char ch1 = '\u0021';

        System.out.println(count);

        System.out.println(floatVal);

        System.out.println(cost);

        System.out.println(hexaVal);

        System.out.println(binary);

        System.out.println(alpha);

        System.out.println(str);

        System.out.println(boolVal);

        System.out.println(octalVal);

        System.out.println(stuName);

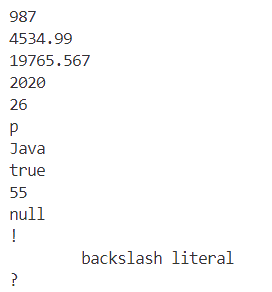
        System.out.println(ch1);

        System.out.println("\t" + "backslash literal");

    }

}

**Output:**

****

**Learning outcome of the Program:**

|  |
| --- |
|  |
|  |
|  |
|  |

**Program 4**

**Program 4:** Write a Java program demonstrating the usage of arithmetic, assignment and unary operators.

**Theory:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Code:**

public class Arithmetic {

    public static void main(String[] args) {

        int res = 11 + 25;

        System.out.println("11 + 25 = " + res);

        int ogRes = res;

        res = res - 5;

        System.out.println(ogRes + " - 5 = " + res);

        ogRes = res;

        res = res \* 7;

        System.out.println(ogRes + " \* 7 = " + res);

        ogRes = res;

        res = res / 2;

        System.out.println(ogRes + " / 2 = " + res);

        ogRes = res;

        res = res % 7;

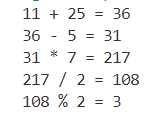
        System.out.println(ogRes + " % 2 = " + res);

        ogRes = res;

    }

}

**Output:**

****

**Learning outcome of the Program:**

|  |
| --- |
|  |
|  |
|  |
|  |

**Program 5**

**Program 5:** Write a Java program to generate random number up to 100 and print whether it is prime number or not.

**Theory:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Code:**

import java.util.Random;

public class prime {

    public static void main(String[] args) {

        Random random=new Random();

        int n=random.nextInt(100);

        System.out.println("random number is "+n);

        boolean isPrime = true;

        for(int i=2;i<=Math.sqrt(n);i++) {

            if(n%i==0) {

                isPrime=false;

                break;

            }

        }

        if(isPrime==true) {

            System.out.println("number is prime");

        } else {

            System.out.println("number is not prime");

        }

    }

}

**Output:**

****

**Learning outcome of the Program:**

|  |
| --- |
|  |
|  |
|  |
|  |

**Program 6**

**Program 6:** (a) Design a Java program to generate first 10 terms of Fibonacci.  
 (b) Find factorial using recursion.

**Theory:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| 1. **Generating First 10 terms of Fibonacci:** |

**Code:**

public class fibo {

    static int n1=0,n2=1,n3=0;

    static void printFibonacci(int count) {

        if(count>0) {

            n3=n1+n2;

            n1=n2;

            n2=n3;

            System.out.println(n3);

            printFibonacci(count-1);

        }

    }

    public static void main(String[] args) {

        int count=10;

        System.out.println(n1);

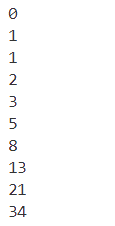
        System.out.println(n2);

        printFibonacci(count-2);

    }

}

**Output:**

****

1. **Finding Factorial using Recursion:**

**Code:**

import java.util.Scanner;

public class Factorial {

    public static void main(String[] args) {

        Scanner input= new Scanner(System.in);

        System.out.print("Enter the number:");

        int num= input.nextInt();

        System.out.println("Factorial of "+ num +" is "+ fact(num));

    }

    static int fact(int num){

        if (num==1 || num==0){

            return 1;

        }

        return num\*fact(num-1);

    }

}

**Output:**

****

**Learning outcome of the Program:**

|  |
| --- |
|  |
|  |
|  |
|  |

**Program 7**

**Program 7:** Design a Java program to find the average sum of array of N numbers entered by user.

**Theory:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Code:**

import java.util.Scanner;

public class SumOfArray {

    public static void main(String[] args) {

        Scanner sin= new Scanner(System.in);

        System.out.print("Enter Number of Elements : ");

        int t=sin.nextInt();

        int[] arr= new int[t];

        for (int i = 0; i < t; i++) {

            System.out.print("Enter Element " + (i+1) + " : ");

            arr[i]=sin.nextInt();

        }

        System.out.println("Sum: "+sum(arr)+"\nAverage: "+average(arr));

    }

    static int sum(int[] arr){

        int res=0;

        for (int x:arr) {

            res+=x;

        }

        return res;

    }

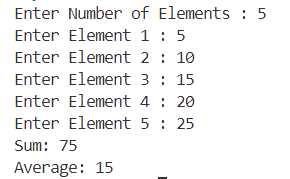
    static int average(int[] arr){

        return sum(arr)/arr.length;

    }

}

**Output:**

****

**Learning outcome of the Program:**

|  |
| --- |
|  |
|  |
|  |
|  |

**Program 8**

**Program 8:** Design a Java program to implement classes and objects  
 (a) Using default constructor.  
 (b) Using parametrized constructor.  
 (c) Using Copy constructor.

**Theory:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Code:**

public class Constructor {

    int x;

    static class DefaultConst{

        int first, second;

    }

    Constructor(int x){

        this.x=x; // Parameter

    }

    Constructor(Constructor old){

        this(old.x); // Clone

    }

    public static void main(String[] args) {

        DefaultConst defaultCon = new DefaultConst();

        Constructor ParamCon = new Constructor(5);

        Constructor CloneCon = new Constructor(ParamCon);

        System.out.print("Default : ");

        System.out.println(defaultCon.first+" "+defaultCon.second);

        System.out.print("Parameterized : ");

        ParamCon.display();

        System.out.print("Copy : ");

        CloneCon.display();

    }

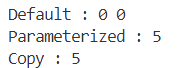
    void display() {

        System.out.println(this.x);

    }

}

**Output:**

****

**Learning outcome of the Program:**

|  |
| --- |
|  |
|  |
|  |
|  |

**Program 9**

**Program 9:** Create a class and find out the area and perimeter of rectangle.

**Theory:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Code:**

import java.util.Scanner;

public class rectangle {

    public static void main(String[] args) {

        Scanner input= new Scanner(System.in);

        System.out.print("Enter length:");

        float len= input.nextFloat();

        System.out.print("Enter breadth:");

        float brd= input.nextFloat();

        System.out.println("Area = " + area(len, brd));

        System.out.println("Perimeter = " + perimeter(len, brd));

    }

    static float area(float x, float y) {

        return x \* y;

    }

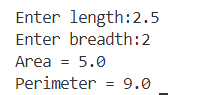
    static float perimeter(float x, float y) {

        return 2\*(x+y);

    }

}

**Output:**

****

**Learning outcome of the Program:**

|  |
| --- |
|  |
|  |
|  |
|  |

**Program 10**

**Program 10:**Create a class circle with instance variable radius and member function   
 (a) area (b) circumference (c) display  
Write a test application named circletest that demonstrate class circled capabilities.

**Theory:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Code:**

1. circle.java

public class circle {

    double areaCalculated;

    double circumferenceCalculated;

    double area(double x) {

        return Math.PI \* x \* x;

    }

    double circumference(double x) {

        return 2 \* Math.PI \* x ;

    }

    public void display() {

        System.out.println("Area = " + areaCalculated);

        System.out.println("Circumference = " + circumferenceCalculated);

    }

}

1. circletest.java

import java.util.Scanner;

public class circletest {

    public static void main(String[] args) {

        circle obj1= new circle();

        Scanner input= new Scanner(System.in);

        System.out.print("Enter the Radius:");

        double rad= input.nextDouble();

        System.out.println("Calculating area...");

        obj1.areaCalculated = obj1.area(rad);

        System.out.println("Calculating circumference...");

        obj1.circumferenceCalculated = obj1.circumference(rad);

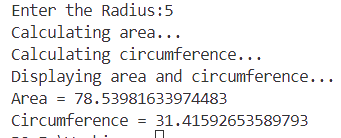
        System.out.println("Displaying area and circumference...");

        obj1.display();

    }

}

**Output:**

****

**Learning outcome of the Program:**

|  |
| --- |
|  |
|  |
|  |
|  |

**Program 11**

**Program 11:** Design a class that perform string operation (equal, reverse and changeCase).

**Theory:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Code:**

import java.util.Scanner;

public class string {

    public static void main(String[] args) {

        Scanner input= new Scanner(System.in);

        System.out.print("Enter String 1 : ");

        String str1= input.nextLine();

        System.out.print("Enter String 2 : ");

        String str2= input.nextLine();

        if (isEqual(str1, str2)) {

            System.out.println("Entered Strings are Equal.");

            System.out.println("String reversed : "+reveString(str1));

            System.out.println("String after changing case : "+changeCase(str1));

        } else {

            System.out.println("Entered Strings are not Equal.");

            System.out.println("String 1 reversed : "+reveString(str1));

            System.out.println("String 2 reversed : "+reveString(str2));

            System.out.println("String 1 after changing case : "+changeCase(str1));

            System.out.println("String 2 after changing case : "+changeCase(str2));

        }

    }

    static boolean isEqual(String x, String y) {

        return x.equals(y);

    }

  static String reveString(String x) {

        char[] charArray = x.toCharArray();

        int left = 0;

        int right = charArray.length - 1;

        while (left < right) {

            char temp = charArray[left];

            charArray[left] = charArray[right];

            charArray[right] = temp;

            left++;

            right--;

        }

        return new String(charArray);

    }

    static String changeCase(String x) {

        char[] charArray = x.toCharArray();

        for (int i = 0; i < charArray.length; i++) {

            if (Character.isLowerCase(charArray[i])) {

                charArray[i] = Character.toUpperCase(charArray[i]);

            } else if (Character.isUpperCase(charArray[i])) {

                charArray[i] = Character.toLowerCase(charArray[i]);

            }

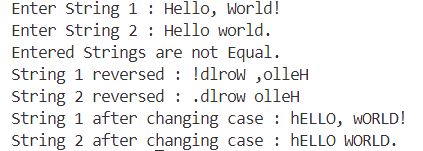
        }

        return new String(charArray);

    }

}

**Output:**

****

**Learning outcome of the Program:**

|  |
| --- |
|  |
|  |
|  |
|  |

**Program 12**

**Program 12:** Write a java program to implement push and pop operation of stack. Also ensure stack overflow and underflow condition are checked while performing push and pop operations.

**Theory:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Code:**

import java.util.Scanner;

public class stack {

    int len;

    int[] stack = {};

    int top = -1;

    Scanner input = new Scanner(System.in);

    public void main(String[] args) {

        System.out.print("Enter lenght of the stack : ");

        this.len = input.nextInt();

        stack = new int[this.len];

        System.out.println("Enter 1 to push\n"+

                        "Enter 2 to pop\n"+

                        "Enter 3 to display\n"+

                        "Enter 4 to exit");

        boolean run = true;

        while (run) {

            System.out.print("Enter Choice (1,2,3,4) : ");

            int ch = input.nextInt();

            switch (ch) {

                case 1:

                    push();

                    break;

                case 2:

                    pop();

                    break;

                case 3:

                    display();

                    break;

                case 4:

                    System.out.println("Exiting...");

                    run = false;

                    break;

                default:

                    System.out.println("Invalid input.");

                    break;

            }

        }

    }

    void push() {

        if (top == len-1) {

            System.out.println("Overflow.");

        } else {

            System.out.print("Enter data : ");

            int data = input.nextInt();

            stack[++top] = data;

        }

    }

    void pop() {

        if (top == -1) {

            System.out.println("Underflow.");

        } else {

            System.out.println(stack[top--] + " popped successfulyy.");

        }

    }

    void display() {

        if (top == -1) {

            System.out.println("No data to show.");

        } else {

            System.out.println("Stack : ");

            for (int i = top; i >= 0; i--) {

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

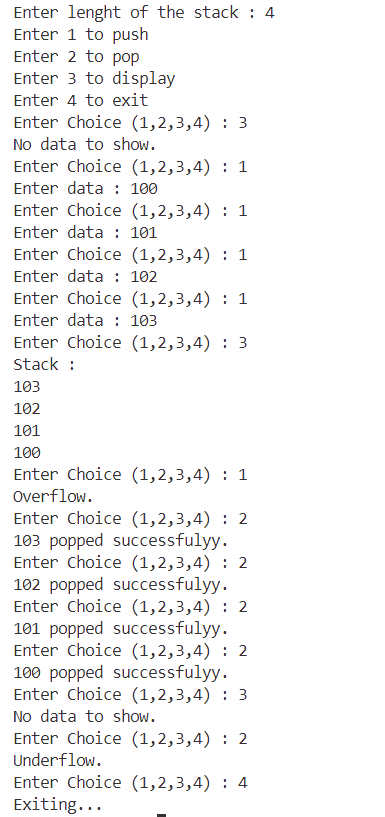
            }

        }

    }

}

**Output:**

****

**Learning outcome of the Program:**

|  |
| --- |
|  |
|  |
|  |
|  |

**Program 13**

**Program 13:** (a) Write a Java program to demonstrate passing object as parameters.  
 (b) Write a Java program to demonstrate the difference between call by value and call by reference.

**Theory:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Code:**

public class Calls {

    int val = 10;

    public static void callByValue(int x) {

        x = 70;

    }

    public static void callByReference(Calls num) {

        num.val = 120;

    }

    public static void main(String[] args) {

        int x = 20;

        Calls num = new Calls();

        System.out.println("Value: " + x);

        callByValue(x);

        System.out.println("Value(After calling the function): " + x);

        System.out.println("Reference: " + num.val);

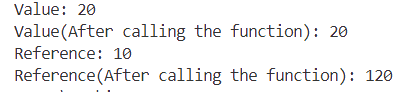
        callByReference(num);

        System.out.println("Reference(After calling the function): " + num.val);

    }

}

**Output:**

****

**Learning outcome of the Program:**

|  |
| --- |
|  |
|  |
|  |
|  |

**Program 14**

**Program 14:** Write a Java program to demonstrate the concept of abstract classes and interfaces.

**Theory:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Code:**

public class Abs {

    public interface Parent {

        void introduce();

    }

    public abstract static class Manners implements Parent {

        abstract void greet();

    }

    public static class Human extends Manners {

        String name;

        public Human(String name) {

            this.name = name;

        }

        void greet() {

            System.out.println("Hello!");

        }

        public void introduce() {

            System.out.println("Myself " + name);

        }

    }

    public static void main(String[] args) {

        Human boy = new Human("Test");

        boy.greet();

        boy.introduce();

    }

}

**Output:**

****

**Learning outcome of the Program:**

|  |
| --- |
|  |
|  |
|  |
|  |

**Program 15**

**Program 15:** Write a Java program to implement inheritance Define a class Box with the following instance variables: width, height and depth, all of type float. Create a new class BoxWeight that extends Box to include weight as an instance variable. Write an application that tests the functionalities of both these classes.

**Theory:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Code:**

import java.util.Scanner;

class Box {

    float width;

    float height;

    float depth;

}

class BoxWeight extends Box {

    float weight;

}

public class inherit {

    public static void main(String[] args) {

        Scanner input= new Scanner(System.in);

        BoxWeight b1 = new BoxWeight();

        System.out.print("Enter width : ");

        b1.width = input.nextFloat();

        System.out.print("Enter height : ");

        b1.height = input.nextFloat();

        System.out.print("Enter depth : ");

        b1.depth = input.nextFloat();

        float volume = b1.width \* b1.height \* b1.depth;

        System.out.println("Volume = " + volume + " unit sq");

        System.out.print("Enter weight : ");

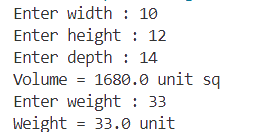
        b1.weight = input.nextFloat();

        System.out.println("Weight = " + b1.weight + " unit");

    }

}

**Output:**

****

**Learning outcome of the Program:**

|  |
| --- |
|  |
|  |
|  |
|  |

**Program 16**

**Program 16:** Implement the following Java programs to demonstrate the concept of exception handling using keywords try, catch, finally, throw and throws wherever required

**Theory:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

(a) Write a Java program using switch to demonstrate the usage of try/catch block for the following handling exceptions:

Case 1: Arithmetic Exception  
 Case 2. Index Out of Bounds Exception  
 Case 3. Null Pointer Exception  
 Case 4. Number Format Exception

**Code:**

import java.util.Scanner;

public class exceptionHandling {

    public static void main(String[] args) {

        Scanner input= new Scanner(System.in);

        System.out.println(

            "1. Arithmetic Exception\n"+

            "2. Index Out of Bound Exception\n"+

            "3. Null Pointer Exception\n"+

            "4. Number Format Exception\n"+

            "5. End"

            );

        int ch;

        boolean run = true;

        while (run) {

            System.out.print("Enter choice (1,2,3,4,5) : ");

            ch = input.nextInt();

            switch (ch) {

                case 1:

                    System.out.println("Arithmetic Exception.");

                    int a = 4;

                    int b = 0;

                    int c;

                    try {

                        c = a/b;

                    } catch (ArithmeticException e) {

                        System.out.println(e);

                    }

                    break;

                case 2:

                    System.out.println("Index Out of Bound Exception.");

                    int[] lst = {};

                    lst = new int[5];

                    try {

                        lst[5] = 10;

                    } catch (IndexOutOfBoundsException e) {

                        System.out.println(e);

                    }

                    break;

                case 3:

                    System.out.println("Null Pointer Exception.");

                    String str = null;

                    try {

                        System.out.println(str.length());

                    } catch (NullPointerException e) {

                        System.out.println(e);

                    }

                    break;

                case 4:

                    System.out.println("Number Format Exception.");

                    try {

                        int charaNum = Integer.parseInt("abc");

                    } catch (NumberFormatException e) {

                        System.out.println(e);

                    }

                    break;

                case 5:

                    System.out.println("Exiting...");

                    run = false;

                    break;

                default:

                    break;

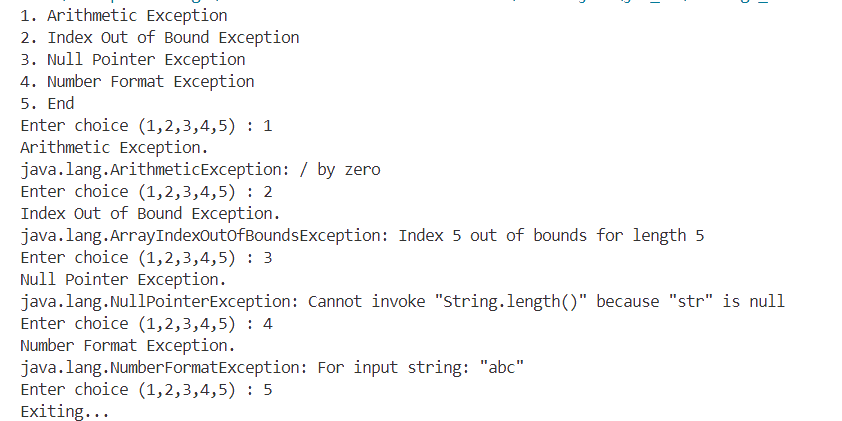
            }

        }

    }

}

**Output:**

****

(b) Write a Java program demonstrate how unreachable code is created and compile-time error occurs when super class exception occurs prior to subclass exception in series of catch statements.

**Code:**

class superClass {

    int a = 5;

    int b = 0;

    int div() {

        return a/b;

    }

}

class subClass extends superClass {

    void appnd() {

        int[] lst = {};

        lst = new int[5];

        lst[5] = 10;

    }

}

public class exceptionHnadligB {

    public static void main(String[] args) {

        subClass cls1 = new subClass();

        try {

            System.out.println("Calling funciton of Super Class...");

            System.out.println(cls1.div());

            System.out.println("Calling funciton of Sub Class...");

            cls1.appnd();

        } catch (ArithmeticException e) {

            System.out.println(e);

        } catch (IndexOutOfBoundsException e) {

            System.out.println(e);

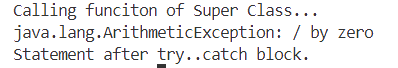
        }

        System.out.println("Statement after try..catch block.");

    }

}

**Output:**

****

(c) Write a Java program that show to catch and handle number format and divison by zero exceptions in programs that use input dialog boxes and/or text fields.

**Code:**

import java.util.Scanner;

public class exceptionHandlingC {

    public static void main(String[] args) {

        Scanner input= new Scanner(System.in);

        boolean run = true;

        while (run) {

            try {

                System.out.print("Enter a : ");

                int a = Integer.parseInt(input.next());

                System.out.print("Enter b : ");

                int b = Integer.parseInt(input.next());

                System.out.println("a = "+a+", b = "+b);

                System.out.println("a / b = "+a/b);

                run = false;

            } catch (NumberFormatException e) {

                System.out.println("Enter Valid Integer...");

            } catch (ArithmeticException e) {

                System.out.print("Arithmetic Error... ");

                System.out.println("Enter valid expression");

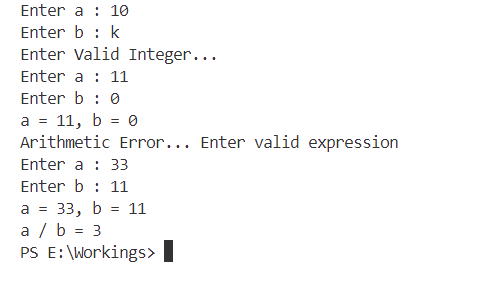
            }

        }

    }

}

**Output:**

****

**Learning outcome of the Program:**

|  |
| --- |
|  |
|  |
|  |
|  |

**Program 17**

**Program 17:** Demonstrate the use of final keyword with data member, function and class.

**Theory:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Code:**

final class FinalClass {

    final int finalVariable;

    FinalClass(int value) {

        this.finalVariable = value;

    }

    final void display() {

        System.out.println("The final variable value is: " + finalVariable);

    }

}

// Attempt to extend the final class would cause a compile-time error

class ExtendedClass extends FinalClass {

    ExtendedClass(int value) {

        super(value);

    }

}

public class useFinal {

    public static void main(String[] args) {

        FinalClass obj = new FinalClass(10);

        obj.display();

        // the following line will cause a compile-time error

        // because the final variable cannot be changed

        obj.finalVariable = 20;

        // this would also result in a compile error

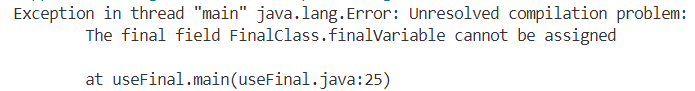
        // because a final class cannot be extended

        ExtendedClass ext = new ExtendedClass(15);

    }

}

**Output:**

** **

**Learning outcome of the Program:**

|  |
| --- |
|  |
|  |
|  |
|  |

**Program 18**

**Program 18:** Write a Java program to demonstrate the usage of following Collections:

List: Array List  
Set  
Map.

**Theory:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Code:**

import java.util.ArrayList;

import java.util.HashSet;

import java.util.HashMap;

import java.util.List;

import java.util.Set;

import java.util.Map;

public class CollectionsDemo {

    public static void main(String[] args) {

        List<String> arrayList = new ArrayList<>();

        arrayList.add("Apple");

        arrayList.add("Banana");

        arrayList.add("Cherry");

        System.out.println("ArrayList: " + arrayList);

        Set<String> hashSet = new HashSet<>();

        hashSet.add("Apple");

        hashSet.add("Banana");

        hashSet.add("Cherry");

        hashSet.add("Apple");

        System.out.println("HashSet: " + hashSet);

        Map<Integer, String> hashMap = new HashMap<>();

        hashMap.put(1, "Apple");

        hashMap.put(2, "Banana");

        hashMap.put(3, "Cherry");

        String itemAtKey2 = hashMap.get(2);

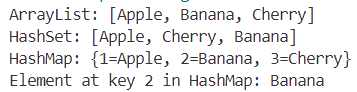
        System.out.println("HashMap: " + hashMap);

        System.out.println("Element at key 2 in HashMap: " + itemAtKey2);

    }

}

**Output:**

****

**Learning outcome of the Program:**

|  |
| --- |
|  |
|  |
|  |
|  |

**Program 19**

**Program 19:** Design a program to demonstrate multi-threading using Thread Class.

**Theory:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Code:**

class NumberThread extends Thread {

    public void run() {

        for (int i = 1; i <= 5; i++) {

            System.out.println("Number: " + i);

            try {

                Thread.sleep(500);

            } catch (InterruptedException e) {

                System.out.println("Number thread interrupted.");

            }

        }

    }

}

class LetterThread extends Thread {

    public void run() {

        char letter = 'A';

        for (int i = 0; i < 5; i++) {

            System.out.println("Letter: " + letter++);

            try {

                Thread.sleep(500);

            } catch (InterruptedException e) {

                System.out.println("Letter thread interrupted.");

            }

        }

    }

}

public class MultiThreadDemo {

    public static void main(String[] args) {

        Thread numberThread = new NumberThread();

        Thread letterThread = new LetterThread();

        numberThread.start();

        letterThread.start();

        try {

            numberThread.join();

            letterThread.join();

        } catch (InterruptedException e) {

            System.out.println("Main thread interrupted.");

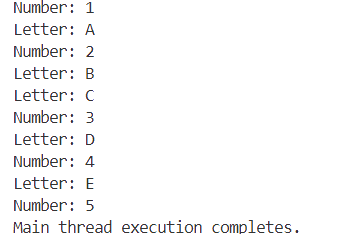
        }

        System.out.println("Main thread execution completes.");

    }

}

**Output:**



**Learning outcome of the Program:**

|  |
| --- |
|  |
|  |
|  |
|  |

**Program 20**

**Program 20:** Design a program to create game ‘Tic Tac Toe’.

**Theory:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Code:**

import java.util.Scanner;

public class TicTacToeGame {

    private static final char EMPTY = '-';

    private static final char PLAYER\_X = 'X';

    private static final char PLAYER\_O = 'O';

    private char[][] board;

    private char currentPlayer;

    private boolean gameEnded;

    public TicTacToeGame() {

        board = new char[3][3];

        for (int i = 0; i < 3; i++) {

            for (int j = 0; j < 3; j++) {

                board[i][j] = EMPTY;

            }

        }

        currentPlayer = PLAYER\_X;

        gameEnded = false;

    }

    public void play() {

        Scanner scanner = new Scanner(System.in);

        while (!gameEnded) {

            printBoard();

            boolean validInput = false;

            while (!validInput) {

                System.out.println(  
 "Player " +

currentPlayer +   
 "'s turn (enter row and column [1-3]):"  
 );

                int row = scanner.nextInt() - 1;

                int col = scanner.nextInt() - 1;

                if (row >= 0 && row < 3 && col >= 0 &&   
 col < 3 && board[row][col] == EMPTY) {

                    board[row][col] = currentPlayer;

                    validInput = true;

                } else {

                    System.out.println(  
 "This move at ("   
 + (row + 1) +   
 "," +   
 (col + 1) + ") is not valid. Try again..."  
 );

                }

            }

            gameEnded = checkForWin() || checkForDraw();

            currentPlayer = (currentPlayer == PLAYER\_X) ? PLAYER\_O : PLAYER\_X;

        }

        scanner.close();

        printBoard();

        if (checkForWin()) {

            System.out.println(  
 "Player " +   
 (currentPlayer == PLAYER\_X ? PLAYER\_O : PLAYER\_X) +   
 " wins!"  
 );

        } else {

            System.out.println("The game ended in a draw!");

        }

    }

    private void printBoard() {

        for (int i = 0; i < 3; i++) {

            for (int j = 0; j < 3; j++) {

                System.out.print(board[i][j]);

                if (j < 2) System.out.print("|");

            }

            System.out.println();

            if (i < 2) System.out.println("-----");

        }

        System.out.println();

    }

    private boolean checkForWin() {

        return (  
 checkRowsForWin() ||   
 checkColumnsForWin() ||   
 checkDiagonalsForWin()  
 );

    }

    private boolean checkRowsForWin() {

        for (int i = 0; i < 3; i++) {

            if (  
 board[i][0] != EMPTY &&   
 board[i][0] == board[i][1] &&   
 board[i][1] == board[i][2]  
 ) {

                return true;

            }

        }

        return false;

    }

    private boolean checkColumnsForWin() {

        for (int j = 0; j < 3; j++) {

            if (board[0][j] != EMPTY &&   
 board[0][j] == board[1][j] &&   
 board[1][j] == board[2][j]) {

                return true;

            }

        }

        return false;

    }

    private boolean checkDiagonalsForWin() {

        return (  
 (  
 board[0][0] != EMPTY &&   
 board[0][0] == board[1][1] &&   
 board[1][1] == board[2][2]  
 ) || (  
 board[0][2] != EMPTY &&   
 board[0][2] == board[1][1] &&   
 board[1][1] == board[2][0]  
 )  
 );

    }

    private boolean checkForDraw() {

        for (int i = 0; i < 3; i++) {

            for (int j = 0; j < 3; j++) {

                if (board[i][j] == EMPTY) {

                    return false;

                }

            }

        }

        return true;

    }

    public static void main(String[] args) {

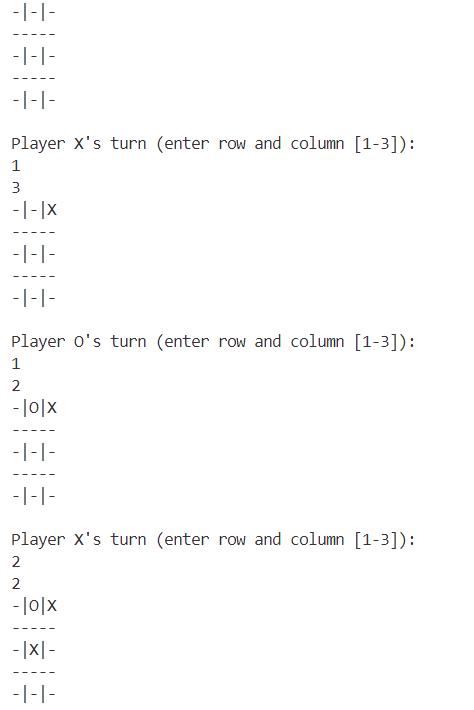
        TicTacToeGame game = new TicTacToeGame();

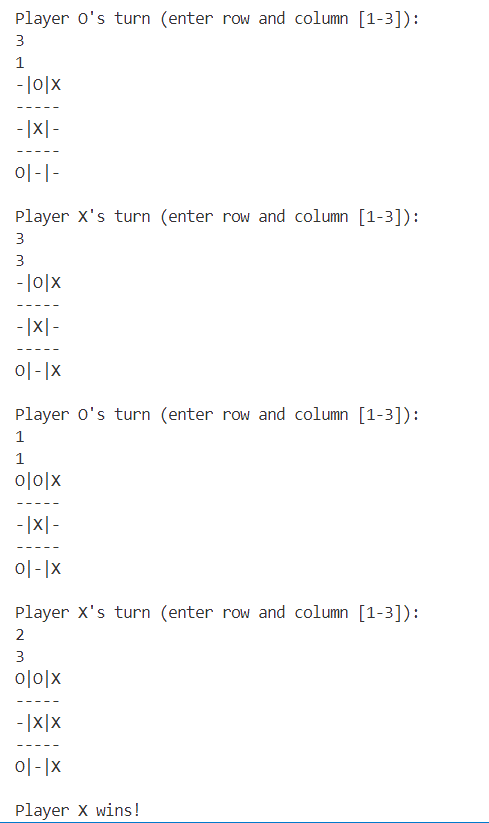
        game.play();

    }

}

**Output:**

****

****

**Learning outcome of the Program:**

|  |
| --- |
|  |
|  |
|  |
|  |

**Program 21**

**Program 21:** Design a program to read a text file and after printing that on screen write the content to another text file.

**Theory:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Code:**

import java.io.\*;

public class FileCopy {

    public static void main(String[] args) {

        String inputFile = "input.txt";

        String outputFile = "output.txt";

        try (BufferedReader reader = new BufferedReader(  
 new FileReader(inputFile)  
 )) {

            String line;

            System.out.println("Content of the input file:");

            while ((line = reader.readLine()) != null) {

                System.out.println(line);

            }

        } catch (IOException e) {

            System.err.println(  
 "Error reading the input file: " +   
 e.getMessage()  
 );

        }

        try (BufferedReader reader = new BufferedReader(  
 new FileReader(inputFile)  
 );

             BufferedWriter writer = new BufferedWriter(  
 new FileWriter(outputFile)  
 )) {

            String line;

            while ((line = reader.readLine()) != null) {

                writer.write(line);

                writer.newLine();

            }

            System.out.println(  
 "\nContent successfully written to the output file."  
 );

        } catch (IOException e) {

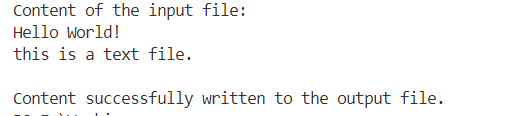
            System.err.println(  
 "Error writing to the output file: " +   
 e.getMessage()  
 );

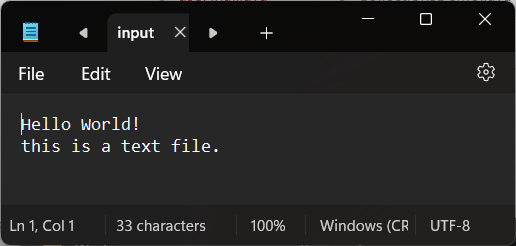
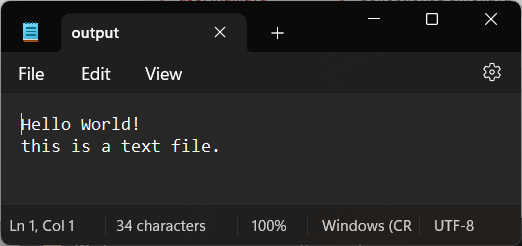
        }

    }

}

**Output:**

****

**Input File: Output File:**

**Learning outcome of the Program:**

|  |
| --- |
|  |
|  |
|  |
|  |

**Program 22**

**Program 22:** Design a program to count number of words, characters, vowels in a text file.

**Theory:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Code:**

import java.io.\*;

public class TextAnalysis {

    public static void main(String[] args) {

        String inputFilePath = "input.txt";

        int characters = 0;

        int words = 0;

        int vowels = 0;

        try (BufferedReader reader = new BufferedReader(  
 new FileReader(inputFilePath)  
 )) {

            String line;

            while ((line = reader.readLine()) != null) {

                characters += line.length();

                String[] wordsArray = line.split("\\s+");

                words += wordsArray.length;

                for (char c : line.toCharArray()) {

                    if (isVowel(c)) {

                        vowels++;

                    }

                }

            }

            System.out.println("Number of characters: " + characters);

            System.out.println("Number of words: " + words);

            System.out.println("Number of vowels: " + vowels);

        } catch (IOException e) {

            System.err.println("Error reading the file: " + e.getMessage());

        }

    }

    private static boolean isVowel(char c) {

        return "AEIOUaeiou".indexOf(c) != -1;

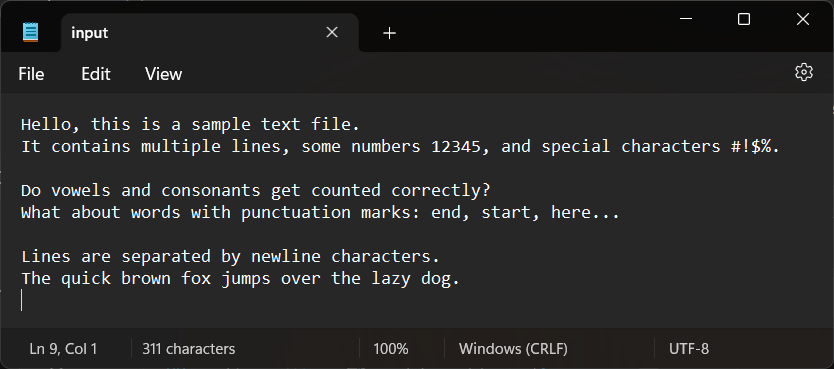
    }

}

**Output:**

****

**Input File:**

****

**Learning outcome of the Program:**

|  |
| --- |
|  |
|  |
|  |
|  |

**Program 23**

**Program 23:** Design a program to create simple chat application using Socket Programming.

**Theory:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Code:**

**ChatServer.java**

import java.io.\*;

import java.net.\*;

public class ChatServer {

    public static void main(String[] args) {

        int port = 1234;

        try (ServerSocket serverSocket = new ServerSocket(port)) {

            System.out.println("Server started. Listening on Port " + port);

            while (true) {

                Socket clientSocket = serverSocket.accept();

                System.out.println("Client connected.");

                Thread handler = new ClientHandler(clientSocket);

                handler.start();

            }

        } catch (IOException e) {

            System.out.println("Server exception: " + e.getMessage());

            e.printStackTrace();

        }

    }

}

class ClientHandler extends Thread {

    private Socket socket;

    public ClientHandler(Socket socket) {

        this.socket = socket;

    }

    public void run() {

        try (BufferedReader reader = new BufferedReader(  
 new InputStreamReader(socket.getInputStream()  
 ));

             PrintWriter writer = new PrintWriter(  
 socket.getOutputStream(), true)) {

            String clientMessage;

            while ((clientMessage = reader.readLine()) != null) {

                System.out.println("Client says: " + clientMessage);

                writer.println("Echo: " + clientMessage);

            }

        } catch (IOException e) {

            System.out.println(  
 "Exception in client handler: " +   
 e.getMessage());

            e.printStackTrace();

        } finally {

            try {

                socket.close();

            } catch (IOException e) {

                e.printStackTrace();

            }

        }

    }

}

**ChatClient.java**

import java.io.\*;

import java.net.\*;

public class ChatClient {

    public static void main(String[] args) {

        String hostname = "localhost";

        int port = 1234;

        try (Socket socket = new Socket(hostname, port)) {

            PrintWriter writer = new PrintWriter(  
 socket.getOutputStream(), true);

            BufferedReader reader = new BufferedReader(new InputStreamReader(  
 socket.getInputStream()));

            BufferedReader consoleReader = new BufferedReader(  
 new InputStreamReader(System.in));

            String userInput;

            System.out.println("Connected to chat server");

            System.out.println("Type a message (or 'exit' to quit):");

while ((userInput = consoleReader.readLine()) != null && !userInput.equalsIgnoreCase("exit")) {

                writer.println(userInput);

                String response = reader.readLine();

                System.out.println(response);

            }

        } catch (UnknownHostException e) {

            System.out.println("Server not found: " + e.getMessage());

        } catch (IOException e) {

            System.out.println("I/O error: " + e.getMessage());

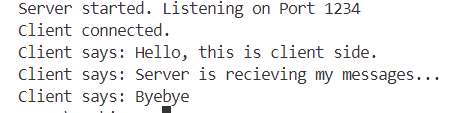
        }

    }

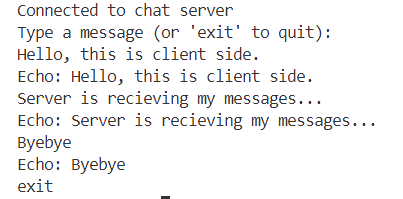
}

**Output:**

**ChatServer.java**

****

**ChatClient.java**

****

**Learning outcome of the Program:**

|  |
| --- |
|  |
|  |
|  |
|  |

**Program 24**

**Program 24:** Design a program to basic calculator using Applet and Event Handling.

**Theory:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Code:**

import java.awt.\*;

import java.applet.\*;

import java.awt.event.\*;

public class Calculator extends Applet implements ActionListener

{

    TextField inp;

    public void init()

    {

    setBackground(Color.white);

    setLayout(null);

    int i;

    inp = new TextField();

    inp.setBounds(150,100,270,50);

    this.add(inp);

    Button button[] = new Button[10];

    for(i=0;i<10;i++)

    {

        button[i] = new Button(String.valueOf(9-i));

        button[i].setBounds(150+((i%3)\*50),150+((i/3)\*50),50,50);

        this.add(button[i]);

        button[i].addActionListener(this);

    }

    Button dec=new Button(".");

    dec.setBounds(200,300,50,50);

    this.add(dec);

    dec.addActionListener(this);

    Button clr=new Button("C");

    clr.setBounds(250,300,50,50);

    this.add(clr);

    clr.addActionListener(this);

    Button operator[] = new Button[5];

    operator[0]=new Button("/");

    operator[1]=new Button("\*");

    operator[2]=new Button("-");

    operator[3]=new Button("+");

    operator[4]=new Button("=");

    for(i=0;i<4;i++)

    {

        operator[i].setBounds(300,150+(i\*50),50,50);

        this.add(operator[i]);

        operator[i].addActionListener(this);

    }

    operator[4].setBounds(350,300,70,50);

    this.add(operator[4]);

    operator[4].addActionListener(this);

    }

    String num1="";

    String op="";

    String num2="";

    //Function to calculate the expression

    public void actionPerformed(ActionEvent e)

    {

    String button = e.getActionCommand();

        char ch = button.charAt(0);

    if(ch>='0' && ch<='9'|| ch=='.')

    {

        if (!op.equals(""))

        num2 = num2 + button;

        else

        num1 = num1 + button;

        inp.setText(num1+op+num2);

    }

    else if(ch=='C')

    {

        num1 = op = num2 = "";

        inp.setText("");

    }

    else if (ch =='=')

    {

        if(!num1.equals("") && !num2.equals(""))

        {

        double temp;

        double n1=Double.parseDouble(num1);

        double n2=Double.parseDouble(num2);

        if(n2==0 && op.equals("/"))

        {

            inp.setText(num1+op+num2+" = Zero Division Error");

            num1 = op = num2 = "";

        }

        else

        {

            if (op.equals("+"))

                temp = n1 + n2;

            else if (op.equals("-"))

                temp = n1 - n2;

            else if (op.equals("/"))

                temp = n1/n2;

            else

                temp = n1\*n2;

            inp.setText(num1+op+num2+" = "+temp);

            num1 = Double.toString(temp);

            op = num2 = "";

            }

            }

        else

        {

        num1 = op = num2 = "";

        inp.setText("");

        }

        }

    else

    {

        if (op.equals("") || num2.equals(""))

        op = button;

        else

        {

        double temp;

        double n1=Double.parseDouble(num1);

        double n2=Double.parseDouble(num2);

        if(n2==0 && op.equals("/"))

        {

            inp.setText(num1+op+num2+" = Zero Division Error");

            num1 = op = num2 = "";

        }

        else

        {

            if (op.equals("+"))

                temp = n1 + n2;

            else if (op.equals("-"))

                temp = n1 - n2;

            else if (op.equals("/"))

                temp = n1/n2;

            else

                temp = n1\*n2;

            num1 = Double.toString(temp);

            op = button;

            num2 = "";

            }

            }

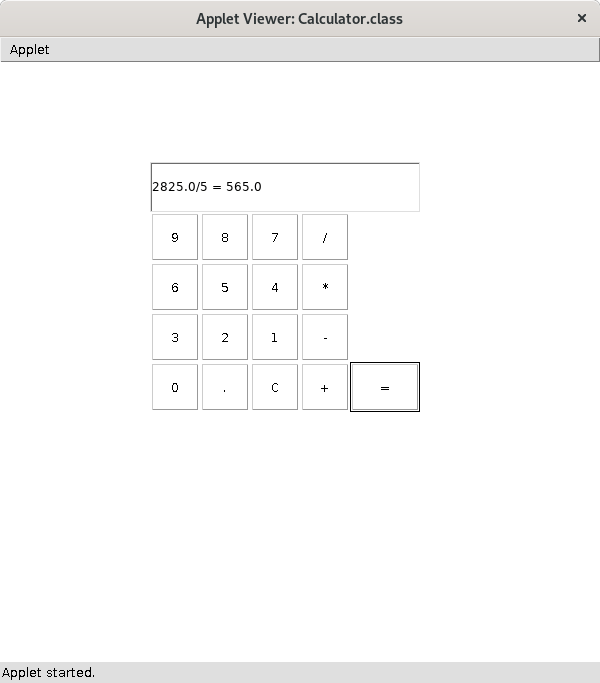
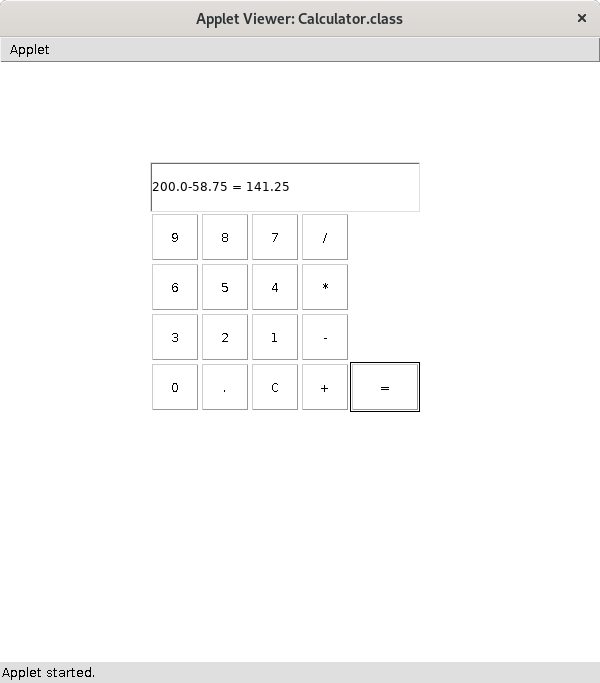
        inp.setText(num1+op+num2);

        }

    }

}

**Output:**



**Learning outcome of the Program:**

|  |
| --- |
|  |
|  |
|  |
|  |

**Program 25**

**Program 25:** Design a program to connect to access database and display contents of the table.

**Theory:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Code:**

import java.sql.\*;

public class AccessDatabaseDemo {

    public static void main(String[] args) {

        String driver = "net.ucanaccess.jdbc.UcanaccessDriver";

        String url = "jdbc:ucanaccess://E:/Workings/sampledb.accdb";

        String username = "";

        String password = "";

        String query = "SELECT \* FROM TableName";

        try {

            Class.forName(driver);

            Connection connection = DriverManager.getConnection(url,   
 username,   
 password  
 );

            Statement statement = connection.createStatement();

            ResultSet resultSet = statement.executeQuery(query);

            while (resultSet.next()) {

                int id = resultSet.getInt("ID");

                String name = resultSet.getString("Name");

                System.out.println("ID: " + id + ", Name: " + name);

            }

            resultSet.close();

            statement.close();

            connection.close();

        } catch (ClassNotFoundException e) {

            System.err.println("Error loading JDBC driver: " +   
 e.getMessage());

        } catch (SQLException e) {

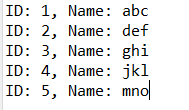
            System.err.println("SQL exception: " + e.getMessage());

        }

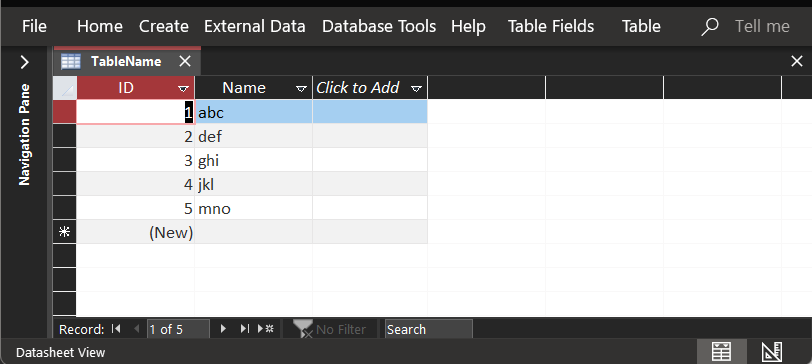
    }

}

**Output:**

****

**MS Access File:**

****

**Learning outcome of the Program:**

|  |
| --- |
|  |
|  |
|  |
|  |