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

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



**Submitted By:**

**Student Name:** Aryan Gupta

**Enrolment no:** 04517711922

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

**Submitted To:**

* Ms. Shikha Jain

**Index**

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Experiment Title | Date | Sign |
| 1. | Getting Familiar with Eclipse: (a) Download and Install Eclipse. (b) Using Eclipse for Java. |  |  |
| 2. | Write a Java program to print “Hello World” to understand compilation and execution of java program. |  |  |
| 3. | Write a Java program demonstrating string concatenation. |  |  |
| 4. | Write java program demonstrating the usage of literal datatypes. |  |  |
| 5. | Write a Java program demonstrating the usage of arithmetic, assignment and unary operators. |  |  |
| 6. | Write a java program demonstrating the usage of pre order and post order operations. |  |  |
| 7. | Write a Java program demonstrating the usage of scanner class for user inputs. |  |  |
| 8. | Write a Java program to demonstrate the usage of Bitwise operators. |  |  |
| 9. | Write a Java program to generate random number up to 100 and print whether it is prime number or not. |  |  |
| 10. | (a) Design a Java program to generate first 10 terms of Fibonacci.  (b) Find factorial using recursion. |  |  |

**Index**

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Experiment Title | Date | Sign |
| 11. | Design a Java program to find the average sum of array of N numbers entered by user. |  |  |
| 12. | Design a Java program to implement classes and objects.  (a) Using default constructor.  (b) Using parametrized constructor.  (c) Using copy constructor. |  |  |
| 13. | Create a class and find out the area and perimeter of rectangle. |  |  |
| 14. | 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 |  |  |
| 15. | Design a class that perform string operation (equal, reverse and changeCase) |  |  |
| 16. | 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. |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Index**

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

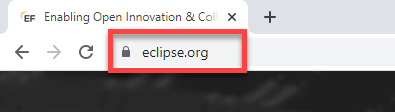
**Experiment 1**

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

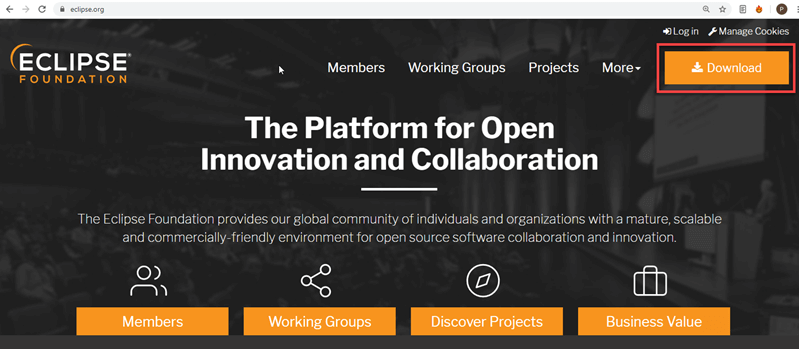
**Eclipse Download and Installation Steps:**

Step 1)**Installing Eclipse**

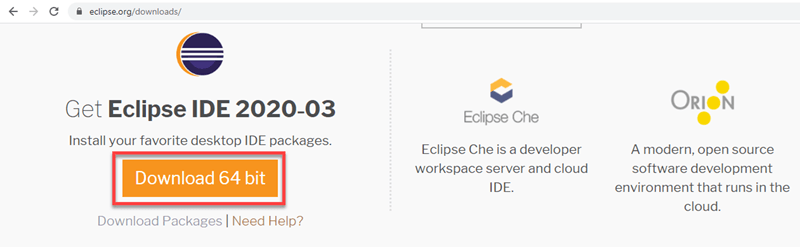
Open your browser and type <https://www.eclipse.org/>

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

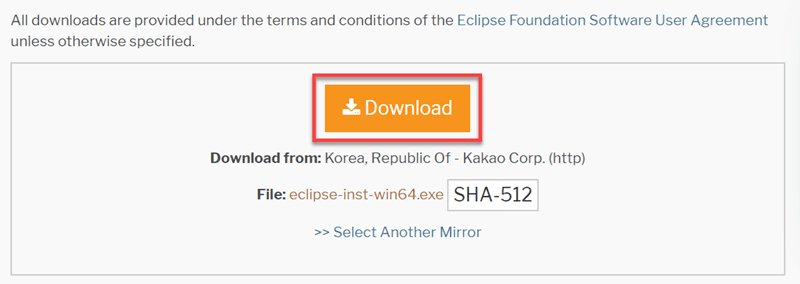
Step 2)**Click on “Download” button.**

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

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

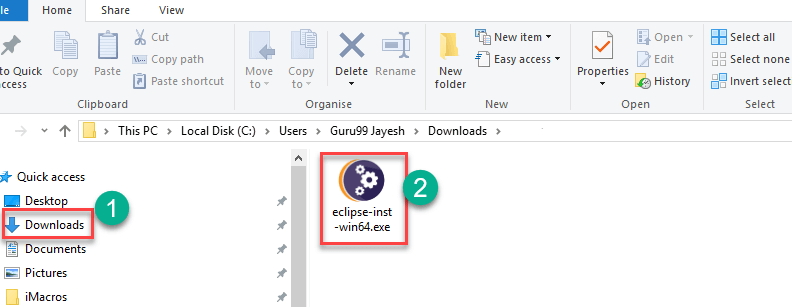
[](https://www.guru99.com/images/2/041720_1109_HowtoDownlo3.png)

Step 4)**Click on “Download” button**

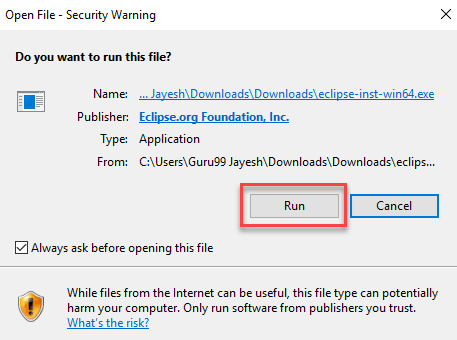
[](https://www.guru99.com/images/2/041720_1109_HowtoDownlo4.png)

Step 4) **Install Eclipse.**

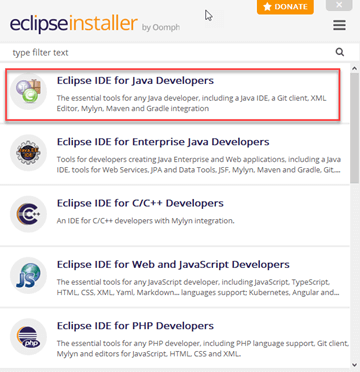
1. Click on “downloads” in Windows file explorer.
2. Click on “eclipse-inst-win64.exe” file.

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

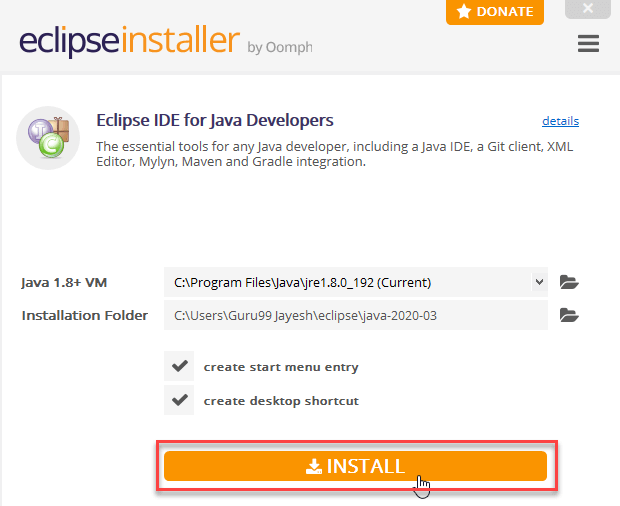
Step 5)**Click on Run button**

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

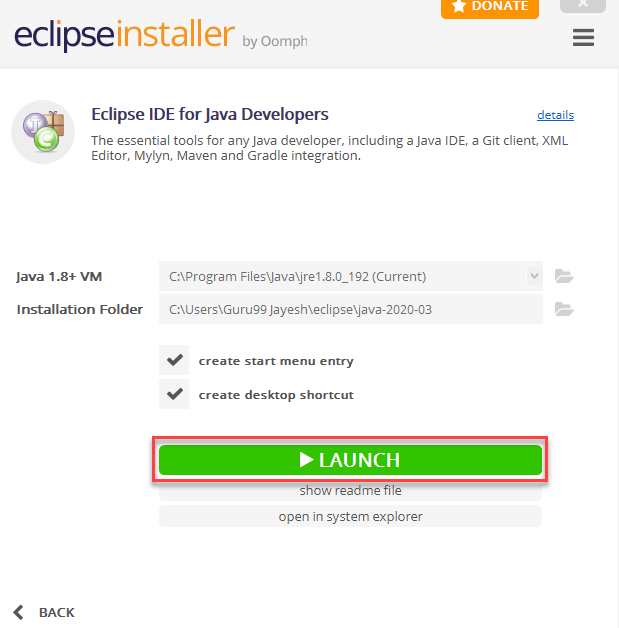
Step 6)**Click on “Eclipse IDE for Java Developers”**

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

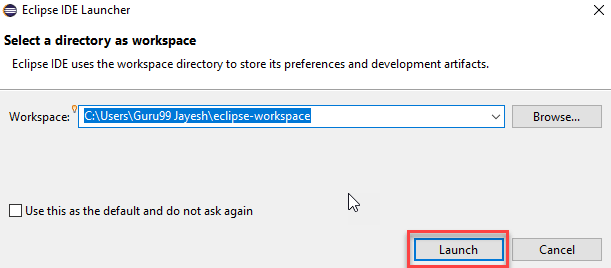
Step 7)**Click on “INSTALL” button**

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

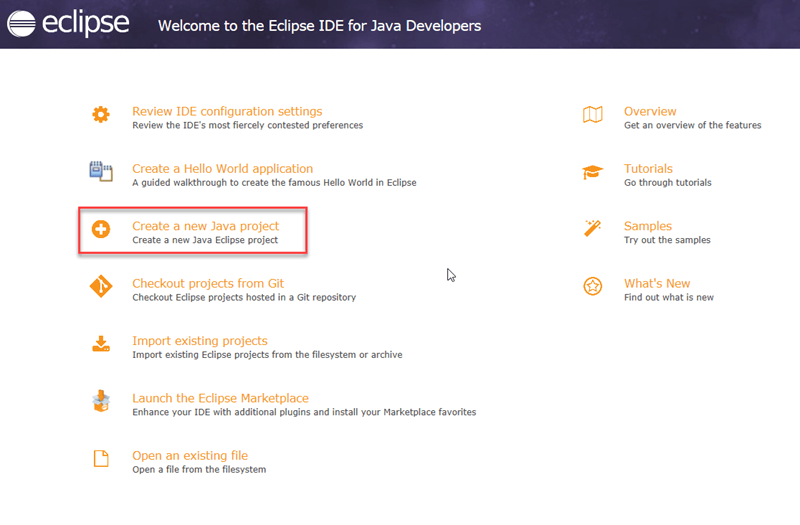
Step 8)**Click on “LAUNCH” button.**

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

Step 9)**Click on “Launch” button.**

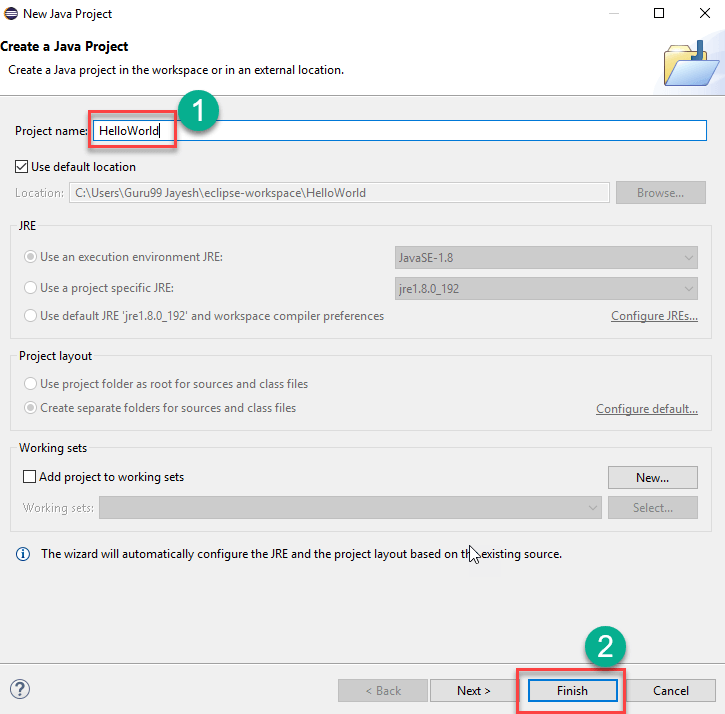
[](https://www.guru99.com/images/2/041720_1109_HowtoDownlo10.png)

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

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

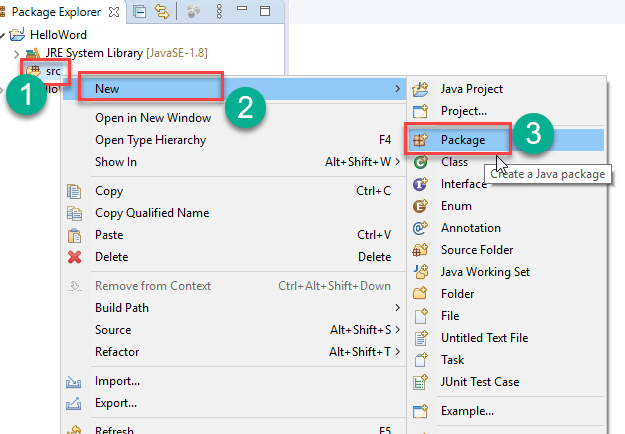
Step 11)**Create a new Java Project**

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

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

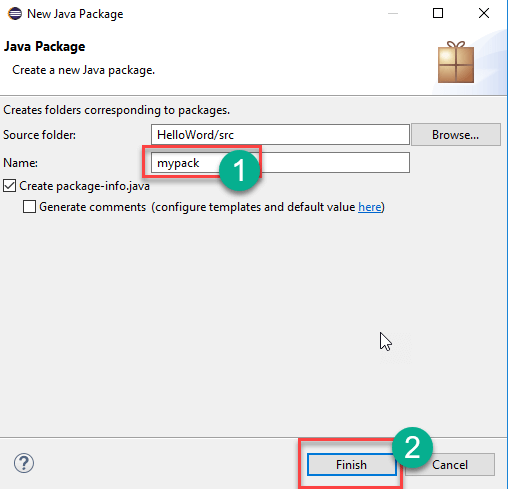
Step 12)[**Create Java Package**](https://www.guru99.com/java-packages.html)**.**

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

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

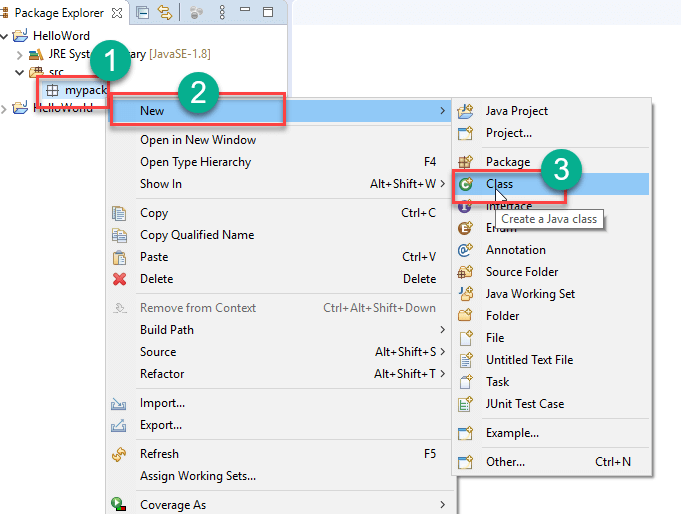
Step 13)**Writing package name.**

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

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

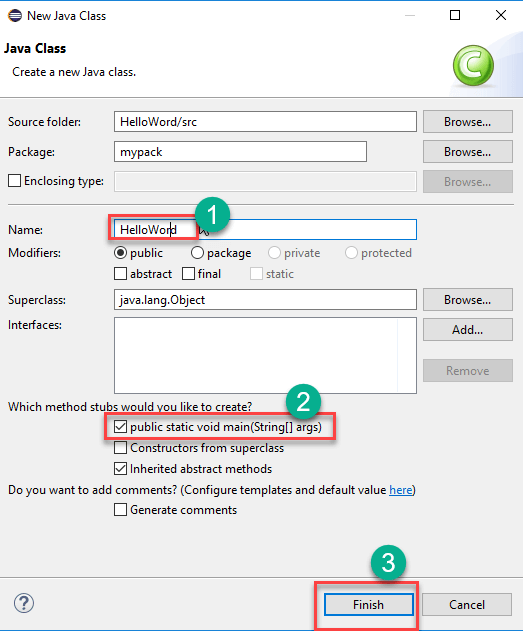
Step 14) **Creating Java Class**

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

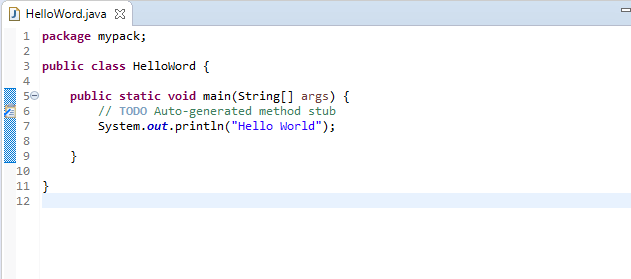
[](https://www.guru99.com/images/2/041720_1109_HowtoDownlo15.png)

Step 15)**Defining Java Class.**

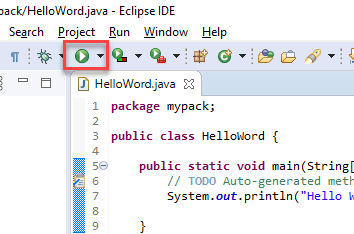
1. Write class name
2. Click on “public static void main (String[] args)” checkbox.
3. Click on “Finish” button.

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

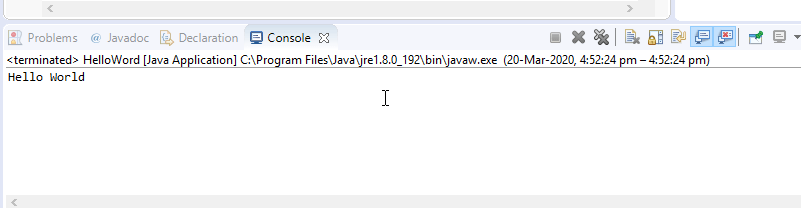
Helloword.java file will be created as shown below:

[](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 Experiment:**

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

**Experiment 2**

**Experiment 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 Experiment:**

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

**Experiment 3**

**Experiment 3:** Write a Java program demonstrating string concatenation.

**Theory:**

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

**Code:**

public class concat {

    public static void main(String[] args) {

        String firstString = "This is";

        String secondString = " a concatenated string.";

        String thirdString = firstString + secondString;

        System.out.println(thirdString);

    }

}

**Output:**

****

**Learning outcome of the Experiment:**

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

**Experiment 4**

**Experiment 4:** 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);

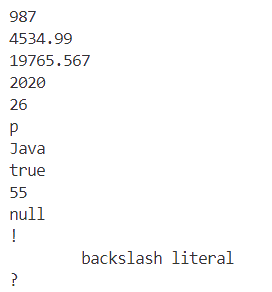
        System.out.println(ch1);

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

    }

}

**Output:**

****

**Learning outcome of the Experiment:**

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

**Experiment 5**

**Experiment 5:** 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;

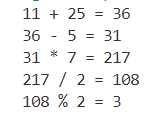
        System.out.println(ogRes + " % 2 = " + res);

        ogRes = res;

    }

}

**Output:**

****

**Learning outcome of the Experiment:**

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

**Experiment 6**

**Experiment 6:** Write a java program demonstrating the usage of pre order and post order operations.

**Theory:**

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

**Code:**

public class PrePost {

    public static void main(String[] args) {

        int i = 7;

        System.out.println("i = " + i);

        System.out.println("++i = " + ++i);

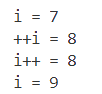
        System.out.println("i++ = " + i++);

        System.out.println("i = " + i);

    }

}

**Output:**

****

**Learning outcome of the Experiment:**

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

**Experiment 7**

**Experiment 7:** Write a Java program demonstrating the usage of scanner class for user inputs.

**Theory:**

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

**Code:**

import java.util.Scanner;

public class scanf {

    public static void main(String[] args) {

        Scanner scanner = new Scanner (System.in);

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

        String name = scanner.nextLine();

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

        String id = scanner.nextLine();

        System.out.println("Student Name = " + name);

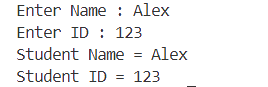
        System.out.println("Student ID = " + id);

        scanner.close();

    }

}

**Output:**

****

**Learning outcome of the Experiment:**

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

**Experiment 8**

**Experiment 8:** Write a Java program to demonstrate the usage of Bitwise operators.

**Theory:**

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

**Code:**

public class bitwise {

    public static void main(String[] args) {

        int a = 5;

        int b = 7;

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

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

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

        System.out.println("~a  = " + (~a));

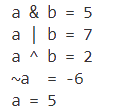
        a &= b;

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

    }

}

**Output:**

****

**Learning outcome of the Experiment:**

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

**Experiment 9**

**Experiment 9:** 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 Experiment:**

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

**Experiment 10**

**Experiment 10:** (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);

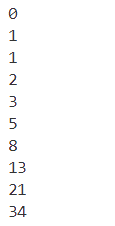
        System.out.println(n2);

        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 Experiment:**

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

**Experiment 11**

**Experiment 11:** 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;

    }

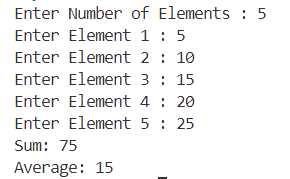
    static int average(int[] arr){

        return sum(arr)/arr.length;

    }

}

**Output:**

****

**Learning outcome of the Experiment:**

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

**Experiment 12**

**Experiment 12:** 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();

    }

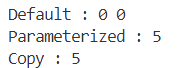
    void display() {

        System.out.println(this.x);

    }

}

**Output:**

****

**Learning outcome of the Experiment:**

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

**Experiment 13**

**Experiment 13:** 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;

    }

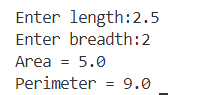
    static float perimeter(float x, float y) {

        return 2\*(x+y);

    }

}

**Output:**

****

**Learning outcome of the Experiment:**

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

**Experiment 14**

**Experiment 14:**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);

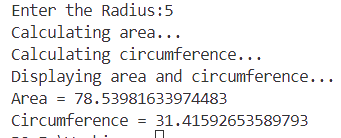
        System.out.println("Displaying area and circumference...");

        obj1.display();

    }

}

**Output:**

****

**Learning outcome of the Experiment:**

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

**Experiment 15**

**Experiment 15:** 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]);

            }

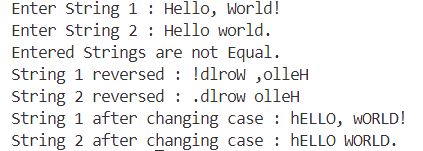
        }

        return new String(charArray);

    }

}

**Output:**

****

**Learning outcome of the Experiment:**

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

**Experiment 16**

**Experiment 16:** 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:**

**Output:**

**Learning outcome of the Experiment:**

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