Experiment No .1

Aim: Write an Installation Step of JDK.

1) Downloading the JDK Installer:

Access the Java SE Downloads page and click **Accept License Agreement**. Under the **Download** menu, click the **Download** link that corresponds to the .exe for your version of Windows.
Download the file jdk-20.*interim.update.patch*_windows-x64_bin.exe.
Note:
Verify the successful completion of file download by comparing the file size on the download page and your local drive. Alternatively, you can ensure that the downloaded file's checksum matches the one provided on the Java SE Downloads page.
Installing the JDK from the JDK .exe Installer
You must have administrator privileges to install the JDK on Microsoft Windows.
To run the JDK installer:

1. Start the JDK 20 installer by double-clicking the installer's icon or file name in the download location.
2. Follow the instructions provided by the installer.
3. After the installation is complete, delete the downloaded file to recover disk space.

Note:
During installation, the following files are copied to your computer:
Copy
"C:\Program Files\Common Files\Oracle\Java\javapath\java.exe"
"C:\Program Files\Common Files\Oracle\Java\javapath\javaw.exe"
"C:\Program Files\Common Files\Oracle\Java\javapath\javac.exe"
"C:\Program Files\Common Files\Oracle\Java\javapath\jshell.exe"
Alternatively, you can run these executables directly from the JDK's bin directory.
Installing the JDK from the MSI Enterprise Installer
The JDK MSI Enterprise Installer is packaged as an .msi installer and runs with minimal dialogs. It supports silent installation of the JDK and is customizable using command line parameters.
You must have administrative permissions install JDK 20.

1. Download jdk-20_windows-x64_bin.msi from the Java SE Downloads page.
2. Double click the .msi file to run the installer.

Installing the JDK from the Command Line
You can install the JDK by downloading the appropriate Oracle JDK MSI Enterprise Installer for your system and running it from the command line.

1. Download jdk-20_windows-x64_bin.msi.
2. Open an MS-DOS prompt with Administrative permissions.
3. Run one of the following commands depending on the type of installation that you want to perform:
    - Basic UI mode:

Copy
msiexec.exe /i jdk-20_windows-x64_bin.msi

- Silent mode:

Copy
msiexec.exe jdk-20_windows-x64_bin.msi /qn

Creating a Log File: You can use a log file to verify that an installation succeeded.

To create a log file describing the installation, append /L C:*path*\setup.log to the install command and scroll to the end of the log file to verify.

The following is an example of creating a log file:

Copy
msiexec.exe /i *installer.msi* /L C:*path*\setup.log

In this example, the log is written to the C:\path\setup.log file.

Installing the JDK Silently

Instead of double-clicking or opening the JDK installer, you can perform a silent, non-interactive, JDK installation by using command-line arguments.

Install JDK in silent mode using the command:

Copy
jdk.exe /s

Note:

- The notation *jdk* stands for the downloaded installer file base name, such as jdk-20_windows-x64_bin.exe.
- You don't need to run the ADDLOCAL command as everything is installed by default.

Beginning to Use the JDK

Select **Java Development Kit** in the Windows **Start** menu to access information related to Reference Documentation.

During JDK installation and uninstallation processes, **Start** menu items are updated so that they are associated with the latest JDK version on the system.

Note:

Windows 10 has a **Start** menu; however, the menu is not available in Windows 8 and Windows 8.1. The JDK and Java information in Windows 8 and Windows 8.1 is available in the following Start directory: %ALLUSERSPROFILE%\Microsoft\Windows\Start Menu\Programs.

Uninstalling the JDK on Windows

To uninstall JDK 20, use the **Add/Remove Programs** utility in the Microsoft Windows Control Panel.

Uninstalling the JDK in Silent Mode

You can use the command line for uninstalling the JDK.

Use the following command to uninstall the JDK in silent mode:

Copy
MsiExec.exe/X{<*UninstallString*>}

For example, to uninstall JDK 18, run the command:

Copy
MsiExec.exe /X{B1405ADC-C0CA-5E63-B1E0-51F5A9A2627C}

To find <*UninstallString*>, see Finding the JDK Registry Key and UninstallString Value.

Note:

- This command can be run from anywhere.
- The msiexec.exe executable is located in the windows system directory.

- A reboot is required only if some files are in use during uninstallation; it is not necessary everytime. However, to manually suppress reboot while uninstalling, append REBOOT=R option to the command.
- Append /l "C:\<*path*>setup.log" option to the command if you want to create a log file describing the uninstallation status.

The **Windows Installer** dialog appears prompting you for confirmation. Click **Yes** to uninstall JDK.

Finding the JDK Registry Key and UninstallString Value

1. Go to the **Start** and type regedit.
2. In the Registry Editor, go to HKEY_LOCAL_MACHINE/Software/Microsoft/Windows/CurrentVersion/Uninstall.

Under the Uninstall folder, you will find many registry entries within curly brackets.

3. Click **Edit** and then **Find**.

Note: Highlight the Uninstall folder before performing search for a particular registry.

4. Enter version string as value to find corresponding registry key. For example, enter **jdk-20**.

The registry key is highlighted on the right-hand side of the pane and values of various uninstall strings are displayed on the left-hand pane.

5. Note the value of the **UninstallString**.

JDK Installation Troubleshooting

The topic provide tips for resolving issues while installing JDK.

- System Error During Decompression
- Program Cannot Be Run in DOS Mode
- Characters That Are Not Part of the System Code Page
- Cleanup the Registry After a Failed JDK Uninstall
- Fixing Shim Situation when JDK 20 and JRE 8 are Installed

System Error During Decompression

If you see the error message: system error during decompression, then there might not be enough space on the disk that contains the TEMP directory.

Program Cannot Be Run in DOS Mode

If you see the error message: This program cannot be run in DOS mode, then do the following:

1. Open the MS-DOS shell or command prompt window.
2. Right-click the title bar.
3. Select **Properties**.
4. Select the **Program** tab.
5. Click **Advanced**.
6. Ensure that the item **Prevent MS-DOS-based programs from detecting Windows** is not selected.
7. Select **OK**.
8. Select **OK** again.
9. Exit the MS-DOS shell.
10. Restart your computer.

Experiment No .2

Aim: Write a Java Program to display Welcome Statement

For executing any Java program, the following software or application must be properly installed.

- Install the JDK if you don't have installed it, download the JDK and install it.
- Set path of the jdk/bin directory. http://www.javatpoint.com/how-to-set-path-in-java
- Create the Java program
- Compile and run the Java program

Program Code:

```
1. class Simple{
2.     public static void main(String args[])
3. {
4.      System.out.println("Wel come to GVAIET");
5.     }
6. }
```

Compilation & Execution commands :

To compile:	javac Simple.java
To execute:	java Simple

Program Output:

Wel come to GVAIET

Experiment No .3

Aim: Write a Java Program to display Arithmetical Operations.

Program Code:

```
mport java.util.Scanner;

public class JavaProgram
{
   public static void main(String args[])
  {
   int first, second, add, subtract, multiply;
   float devide;
   Scanner scanner = new Scanner(System.in);

   System.out.print("Enter Two Numbers : ");
   first = scanner.nextInt();
   second = scanner.nextInt();

   add = first + second;
   subtract = first - second;
   multiply = first * second;
   devide = (float) first / second;

   System.out.println("Sum = " + add);
   System.out.println("Difference = " + subtract);
   System.out.println("Multiplication = " + multiply);
   System.out.println("Division = " + devide);
  }
}
```

Program Output:

```
Enter Two Numbers : 12
5

Sum = 17
Difference = 7
Multiplication = 60
Division = 2.4
```

Experiment No .4

Aim: Write a Java Program to find Area of Circle.

Program Code:

```
1. /*
2.  * Java program to find the area of a circle
3.  */
4.
5. public class Circle
6. {
7.     public static void main(String[] args)
8.     {
9.         int r;
10.        double pi = 3.14, area;
11.        Scanner s = new Scanner(System.in);
12.        System.out.print("Enter radius of circle: ");
13.        r = s.nextInt();
14.        area = pi * r * r;
15.        System.out.println("Area of circle: "+area);
16.    }
17.}
```

Program Output:

```
$ javac Circle.java
$ java Circle
Enter radius of circle: 22
Area of circle: 1519.76
```

Experiment No .5

Aim: Write a Java Program to display Arithmetical Operations using switch case.

Program Code:

```
import java.util.Scanner;

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

    char operator;
    Double number1, number2, result;

    // create an object of Scanner class
    Scanner input = new Scanner(System.in);

    // ask users to enter operator
    System.out.println("Choose an operator: +, -, *, or /");
    operator = input.next().charAt(0);

    // ask users to enter numbers
    System.out.println("Enter first number");
    number1 = input.nextDouble();

    System.out.println("Enter second number");
    number2 = input.nextDouble();

    switch (operator) {

      // performs addition between numbers
      case '+':
        result = number1 + number2;
        System.out.println(number1 + " + " + number2 + " = " + result);
        break;

      // performs subtraction between numbers
      case '-':
        result = number1 - number2;
        System.out.println(number1 + " - " + number2 + " = " + result);
        break;

      // performs multiplication between numbers
      case '*':
        result = number1 * number2;
        System.out.println(number1 + " * " + number2 + " = " + result);
        break;
```

```
    // performs division between numbers
    case '/':
      result = number1 / number2;
      System.out.println(number1 + " / " + number2 + " = " + result);
      break;

    default:
      System.out.println("Invalid operator!");
      break;
  }

  input.close();
 }
}
```

Program Output:

```
Choose an operator: +, -, *, or /
*
Enter first number
3
Enter second number
9
3.0 * 9.0 = 27.
```

Experiment No .6

Aim: Write a Java Program to display 1 to N Numbers.

Program Code:

```
class JavaExample {
  public static void main(String args[]) {
        int n = 30;
        System.out.print("Numbers  from 1 to "+n+" are: ");
        for (int i = 1; i <= n; i++) {
              System.out.print(i + " ");
        }
  }
}
```

Program Output :

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Experiment No .7

Aim: Write a Java Program for Command Line Argument.

Program Code:

```
public class Arithmetic {

  public static void main(String[ ] args) {
    int a = Integer.parseInt(args[0]);
    int b = Integer.parseInt(args[1]);
    int sum  = a + b;
    int prod = a * b;
    int quot = a / b;
    int rem  = a % b;
    int sub  = a - b;

    System.out.println(a + " + " + b + " = " + sum);
    System.out.println(a + " * " + b + " = " + prod);
    System.out.println(a + " / " + b + " = " + quot);
    System.out.println(a + " - " + b + " = " + sub);
    System.out.println(a + " % " + b + " = " + rem);
    System.out.println(a+ " = " +quot+ " * " +b+ " + " +rem);
  }
}
```

Program Output:

```
10 + 6 = 16
10 * 6 = 60
10 / 6 = 1
10 - 6 = 4
10 % 6 = 4
10 = 1 * 6 + 4
```

Experiment No .8

Aim: Write a Java Program for to create Object Using New Keyword.

Program Code:

```
public class Rectangle {

 int height, width;

 Rectangle(int height, int width) {
  this.height = height;
  this.width = width;
 }
 void displayArea() {
  System.out.println("Area of the rectangle is: " + height * width + " sq. units");
 }
 public static void main(String[] args) {
  Rectangle obj = new Rectangle(60, 30);
  obj.displayArea();
 }
}
```

Program Output:

```
Area of the rectangle is: 1800 sq. units
```

Experiment No .9

Aim: Write a Java Program for to create Methods to collect Employee Data.

Program Code:

```
1. package JavaTpoint.JavaObjectToJSON;
2. //Creating Employee class
3. class EmployeeDetails {
4.     //Creating properties of Employee class
5. int emp_id, salary;
6. String name, address, department, email;
7.
8. //Getter and setters for getting and setting properties
9. public int getEmp_id() {
10.    return emp_id;
11. }
12. public void setEmp_id(int emp_id) {
13.    this.emp_id = emp_id;
14. }
15. public int getSalary() {
16.    return salary;
17. }
18. public void setSalary(int salary) {
19.    this.salary = salary;
20. }
21. public String getName() {
22.    return name;
23. }
24. public void setName(String name) {
25.    this.name = name;
26. }
```

```
27. public String getAddress() {
28.     return address;
29. }
30. public void setAddress(String address) {
31.     this.address = address;
32. }
33. public String getDepartment() {
34.     return department;
35. }
36. public void setDepartment(String department) {
37.     this.department = department;
38. }
39. public String getEmail() {
40.     return email;
41. }
42. public void setEmail(String email) {
43.     this.email = email;
44. }
45.
46. //Overriding toString() method
47. @Override
48. public String toString() {
49.     return "Employee [emp_id = " + emp_id + ", salary = " + salary + ", name = " +
    name + ", address = " + address
50.             + ", department = " + department + ", email = " + email + "]";
51.     }
52.
53. }
54. //Creating main class
```

```
55. public class Employee{
56.    //main() method start
57.    public static void main(String args[]) {
58.
59.       //Creating object of EmployeeDetails class
60.       EmployeeDetails emp = new EmployeeDetails();
61.       //Setting values to the properties
62.       emp.setEmp_id(101);
63.       emp.setName("Emma Watson");
64.       emp.setDepartment("IT");
65.       emp.setSalary(15000);
66.       emp.setAddress("New Delhi");
67.       emp.setEmail("Emmawatson123@gmail.com");
68.
69.       //Showing Employee details
70.       System.out.println(emp);
71.
72.       //Getting salary using getter
73.       int sal = emp.getSalary();
74.       int increment = 0;
75.       //Incrementing salary based on condition
76.       if ((sal >=1000) && (sal <=1500))
77.       {
78.          //incrementing salary 2%
79.          increment += (sal * 2)/100;
80.          sal = sal+increment;
81.
82.          emp.setSalary(sal);
83.          System.out.println("\n Salary is incremented \n");
```

```
84.             System.out.println(emp);
85.
86.         }else if ((sal >=1500) && (sal <=20000)){
87.             //incrementing salary 5%
88.             increment += (sal * 5)/100;
89.             sal = sal+increment;
90.
91.             emp.setSalary(sal);
92.             System.out.println("\n Salary is incremented \n");
93.             System.out.println(emp);
94.         }else {
95.             System.out.println("\n Salary is not incremented \n");
96.             System.out.println(emp);
97.         }
98.     }
99. }
```

Program Output:

Employee [emp_id = 101, salary = 15000, name = Emma Watson, address = New Delhi, department = IT, email = Emmawatson123@gmail.com]

Salary is incremented

Employee [emp_id = 101, salary = 15750, name = Emma Watson, address = New Delhi, department = IT, email = Emmawatson123@gmail.com]

Experiment No .10

Aim:Write a Java Program using super and final keyword

Program Code:

```
class Vehicle {
   final String
   type;

   public Vehicle(String
      type) {this.type =
      type;
   }
   public void displayInfo() {
      System.out.println("This is a " + type + " vehicle.");
   }
}
class Car extends Vehicle
   {private String brand;

   public Car(String type, String
      brand) {super(type);
      this.brand = brand;
   }
   public void
      displayInfo() {
      super.displayInfo();
      System.out.println("It is a " + brand + " car.");
   }
}
public class Main {
   public static void main(String[] args) {
      Car myCar = new Car("Four-wheeler", "Toyota"
      myCar.displayInfo();

   }

}
```

Experiment No .11

Aim: Write a Java Program on exception handling

Program Code:

```
public class
   ExceptionHandlingDemo {
   public static void main(String[]
   args) {
      try {
         int numerator =
         10; int
         denominator = 0;
         int result = numerator / denominator; // This will throw an
         ArithmeticExceptionSystem.out.println("Result: " + result);
      } catch (ArithmeticException e) {
         System.err.println("An arithmetic exception occurred: " + e.getMessage());
      } finally {
         System.out.println("This code block will always be executed, even if an
exceptionoccurs.");
      }

      try {
         String str = null;
         System.out.println(str.length()); // This will throw a NullPointerException
      } catch (NullPointerException e) {
         System.err.println("A null pointer exception occurred: " + e.getMessage());
      }

      try {
         int[] numbers = new int[3];
         numbers[5] = 10; // This will throw an ArrayIndexOutOfBoundsException
      } catch (ArrayIndexOutOfBoundsException e) {
         System.err.println("An array index out of bounds exception occurred: " +
         e.getMessage());
```

```
        }

        System.out.println("Program continues to execute after exception handling.");

    }

}
```

Experiment No .12

Aim: Write a Java Program on user defined exception

Program Code:

```
class AgeOutOfRangeException extends
   Exception { public
   AgeOutOfRangeException(String message) {
      super(message);
   }
}

class Person {
   private String
   name;private int
   age;

   public Person(String name, int age) throws AgeOutOfRangeException
      {this.name = name;
      if (age < 0 || age > 120) {
         throw new AgeOutOfRangeException("Invalid age: " + age);
      }
      this.age = age;
   }

   public void displayInfo() {
      System.out.println("Name: " +
      name);System.out.println("Age:
      " + age);
   }
}
```

```
public class
   UserDefinedExceptionDemo {
   public static void main(String[]
   args) {
      try {
         Person person1 = new Person("Alice",
         25);person1.displayInfo();

         // Attempt to create a person with an
         invalid agePerson person2 = new
         Person("Bob", 150);
         person2.displayInfo(); // This won't be reached due to the exception
      } catch (AgeOutOfRangeException e) {
         System.err.println("Error: " + e.getMessage());
      }
   }
}
```

Experiment No .13

Aim: Write a Java Program on multithreading

Program Code:

```
class MyRunnable implements
   Runnable {public void run() {
      for (int i = 1; i <= 5; i++) {
         System.out.println("Thread " + Thread.currentThread().getId() + "
         Count: " + i);try {
            Thread.sleep(1000); // Sleep for 1 second
         } catch
            (InterruptedException e) {
            System.out.println(e);
         }
      }
   }
}

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

```
        MyRunnable myRunnable = new MyRunnable();

        Thread thread1 = new
        Thread(myRunnable);Thread thread2
        = new Thread(myRunnable);

        thread1.start(); // Start the first thread
        thread2.start(); // Start the second thread
    }
}
```

Experiment No .14

Aim: Write a Java Program on graphics class

Program Code:

```
import java.awt.Frame;
import java.awt.Graphics;
import
java.awt.event.WindowAdapter;
import
java.awt.event.WindowEvent;

public class GraphicsDemo extends
   Frame {public GraphicsDemo() {
      setSize(400, 400);
      setTitle("Graphics Demo");
      addWindowListener(new
         WindowAdapter() { public void
         windowClosing(WindowEvent we) {
            System.exit(0);
         }
      });
      setVisible(true);
   }
   public void paint(Graphics g) {
      g.drawRect(100, 100, 200, 150); // Draw a rectangle (x, y, width, height)
   }

   public static void main(String[] args) {
      GraphicsDemo demo = new GraphicsDemo();
   }

}
```

Experiment No .15

Aim:Write a Java Program on Strings

Program Code:

```
public class StringOperations {
    public static void main(String[] args) {
        // Creating strings
        String str1 =
        "Hello, ";String
        str2 = "world!";

        // Concatenation
        String concatenatedString = str1 + str2;
        System.out.println("Concatenated String: " + concatenatedString);

        // Length of a string
        int length = concatenatedString.length();
        System.out.println("Length of the String: " + length);

        // Accessing characters in a string
        char firstChar = concatenatedString.charAt(0);
        char lastChar =
        concatenatedString.charAt(length - 1);
        System.out.println("First Character: " +
        firstChar); System.out.println("Last Character:
        " + lastChar);

        // Substring
        String substring = concatenatedString.substring(7, 12); // "world"
        System.out.println("Substring: " + substring);
```

```
        // String comparison
        String str3 = "Hello, World!";
        boolean isEqual =
        concatenatedString.equals(str3);
        System.out.println("String Comparison: " +
        isEqual);

        // Case conversions

        String lowercaseString =
        concatenatedString.toLowerCase(); String
        uppercaseString = concatenatedString.toUpperCase();
        System.out.println("Lowercase: " + lowercaseString);
        System.out.println("Uppercase: " + uppercaseString);

        // Searching for a substring
        boolean containsWorld =
        concatenatedString.contains("world");
        System.out.println("Contains 'world': " + containsWorld);

        // Replacing characters
        String replacedString = concatenatedString.replace('o', '0');
        System.out.println("String after replacing 'o' with '0': " + replacedString);

        // Splitting a string
        String[] words = concatenatedString.split("
        ");System.out.println("Words in the
        string:"); for (String word : words) {
            System.out.println(word);
        }
    }
}
```

Experiment No .16

Aim: Write a Java Program on arrays

Program Code:

```
public class MatrixMultiplication {
    public static void main(String[] args) {
        // Define two matrices
        int[][] matrix1 = {
            {1, 2, 3},
            {4, 5, 6},
        };

        int[][] matrix2 = {
            {7, 8},
            {9, 10},
            {11, 12},
        };

        // Check if multiplication is possible
        int columns1 = matrix1[0].length; // Number of columns in the first
        matrixint rows2 = matrix2.length;      // Number of rows in the
        second matrix

        if (columns1 != rows2) {
            System.out.println("Matrix multiplication is not possible.");
            return;
        }
```

```
    // Create the result matrix
    int rows1 = matrix1.length;     // Number of rows in the first matrix
    int columns2 = matrix2[0].length; // Number of columns in the second
    matrixint[][] resultMatrix = new int[rows1][columns2];

    // Perform matrix multiplication
    for (int i = 0; i < rows1; i++) {
      for (int j = 0; j < columns2; j++) {
        int sum = 0;
        for (int k = 0; k < columns1; k++) {
          sum += matrix1[i][k] * matrix2[k][j];
        }
        resultMatrix[i][j] = sum;
      }
    }

    // Print the result
    System.out.println("Result Matrix (Matrix 1 * Matrix 2):");
    for (int i = 0; i < rows1; i++) {
      for (int j = 0; j < columns2; j++) {
        System.out.print(resultMatrix[i][j] + " ");
      }
      System.out.println();
    }
  }
}
```

Experiment No .17

Aim: Write a Java Program on applet

Program Code:

SimpleApplet.java

```
import java.applet.Applet;
import java.awt.Graphics;

public class SimpleApplet extends Applet
  {public void paint(Graphics g) {
     g.drawString("Hello, World!", 50, 25);
  }
}
```

Applet.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Simple Applet Example</title>
</head>
<body>
  <applet code="SimpleApplet.class" width="200" height="50"></applet>
</body>
</html>
```