PIRENS Institute of Business Management and Administration, Loni BK.		
Roll Number:	**Date: / /**	**Sign:**
Student Name:		
Subject Name: IT11 JAVA PROGRAMMING		
Program Title: Q1. Installation of jdk environment & following utilities. What is javac, javap and javadoc.		

Q1. Installation of JDK environment & following utilities. What is javac, javap and javadoc.

The Java Development Kit (JDK) is helpful to write programs in Java. It is also known as the Java Platform Standard Edition (Java SE). JDK is an open-source software provided by Sun Microsystems, now taken over by Oracle, for anyone to use freely for programming

Do many people have the question, JDK or JRE? So let us look at what each is used for. Java Runtime Environment (JRE) is required for running programs of Java, whereas JDK is required for writing and running the programs. JDK is a development kit containing JRE and the development tools (debugger and compiler) needed for getting a program output in Java. Simply put, JRE happens to be a subset of JDK; that is, it is included in JDK. Therefore, we will need to install JDK to write and run programs.

Installation Prerequisites of JDK

JDK has bare minimum requirements for disk space and RAM for the 64-bit Windows platform. It requires around 800 MB disk space to install JDK, as JRE also gets installed along with it. JDK requires 128 MB of memory space to run JDK successfully. This is the minimum RAM required for running basic and small programs, but as the size of an application increases, the memory requirement also increases for the application to run smoothly.

Step by Step Installation Of JDK

Before installing the latest JDK version, it is recommended to check our systems for any old JDK versions and uninstall them. Though we can have more than one JDK, it is easier to set paths and to work with just the latest one.

Step 1: Download JDK from the Site

- Go to the Oracle site and open the Java SE download page. Under the latest version of Java Platform, Standard Edition, click on the JDK download button.

  Next, click on the Accept License Agreement button and choose your version of Java for Windows (32-bit or 64-bit) to proceed with downloading the JDK executable file.

  Step 2: Install the JDK exe File

- In this step, we will be running the executable JDK file (It will be a file with .exe as an extension) once the download is done. This installs JDK as well as JRE. For running this file on Windows, we will need Administrator rights.

- To begin the installation, we need to double-click on the downloaded file, and we will be presented with the below window.

- Click on Next to proceed with the installation.
- Click on the Close button once the installation has finished.
- To recover some of our system's disk space, it is good practice to delete the downloaded exe file once the download has been done.

Step 3: Check the Directory

- JDK gets installed in the C directory of our system by default having the path "C:\Program Files\Java\jdk-11.0". If we make any change to this path at all, we need to make a note of itas it will be required in the upcoming steps.
- This is the directory structure for our example.

Step 4: Update the Environment Variables

- We will need to update our system's Environment variables with our installed JDK bin path to run the Java programs because while executing the programs, the command promptwill look for the complete JDK bin path.
- The PATH variable in our system provides the exact location of executables that will be used for running Java programs, such as javac and java. The CLASSPATH variable provides us with the library files location.
- If we do not set the PATH variable, we will specify the full path to the JDK bin every time we run a program.

For example: C:\> "C:\Program Files\Java\jdk-11.0\bin\javac" TestClass.java

- So to set these variables, first right-click on My PC and select Properties.
- Inside Properties, in the left-side panel, select Advanced System Settings, and here choosethe option Environment Variables.
- Click on New, and type PATH in the Variable Name, and enter the path of the bin of installed JDK in the Variable Value field.
- If we already have the PATH variable, we can edit it by adding it to the existing values.
- Click on the OK button to apply the changes.

Step 5: Verify the Java Installation

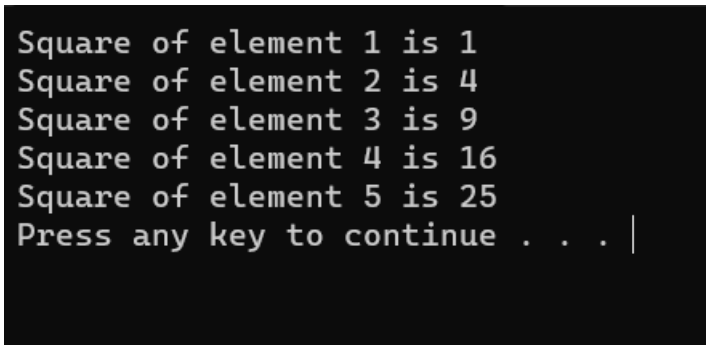
- Open the command prompt and enter the command "java –version", and if it runs successfully, Java has been successfully installed.
- Now that we have seen the steps to install JDK, let the programming fun begin!

PIRENS Institute of Business Management and Administration, Loni BK.
Roll Number: **Date: / /** **Sign:**
Student Name:
Subject Name: IT11 JAVA PROGRAMMING
Program Title: Q 2 Design an application by using array

Program:

```
public class LookupTableUsingArray{

    public static void main(String args[]) {

        // defining the lookup table using array
        int[] lookupTable = new int[5];

        // calculating and storing the square of elements
        // directly into the lookup table
        for(int i = 0; i < 5; i++){

            lookupTable[i] = (i+1)*(i+1);
        }


        // displaying the sqaure values
        for(int j = 0; j < 5; j++){
            System.out.println("Square of element " +(j+1)+" is " + lookupTable[j]);
        }

    }

}
```

Output:



PIRENS Institute of Business Management and Administration, Loni BK.		
Roll Number:	**Date: / /**	**Sign:**
Student Name:		
Subject Name: IT11 JAVA PROGRAMMING		
Program Title: Q.3 Implementation of package, Interface and abstract class		

Program :

```
// Implementation of Interface using Abstract Class in Java

// Interface GFG
interface GFG {
	void learnCoding();
	void learnProgrammingLanguage();
	void contribute();
}

// Abstract class Student implementing from GFG interface
abstract class Student implements GFG {


	@Override public void learnCoding()  // Overriding the methods
	{
		System.out.println("Let's make coding a habit with GFG");
	}
	@Override public void learnProgrammingLanguage()
	{
		System.out.println("Let's master all fundamentals of java with the help of GFG");
	}
}

class GEEK extends Student   // Extend the GEEK class by Student abstract class
{
	@Override public void contribute()
	{
		System.out.println("Now let's help others by contributing in GFG");
	}
}

public class Main   // Driver code
{
	public static void main(String[] args)
	{
		// New GEEK object is created
		GEEK acbstud = new GEEK();

		// Calls to the multiple functions
		acbstud.learnCoding();
		acbstud.learnProgrammingLanguage();
		acbstud.contribute();
	}
}
```

Output:

```
C:\Users\dhote\OneDrive\Documents>java Main
Let's make coding a habit with GFG
Let's master all fundamentals of java with the help of GFG
Now let's help others by contributing in GFG
```

PIRENS Institute of Business Management and Administration, Loni BK.		
Roll Number:	**Date: / /**	**Sign:**
Student Name:		
Subject Name: IT11 JAVA PROGRAMMING		
Program Title: Q.4 Design application using String, StringBuilder, StringTokenizer		

A) Using String,StringBuilder:

Program:

```
import java.util.*;
public class stringlength
{
 public static void main(String args[])
        {

        StringBuilder sb=new StringBuilder("jony");

         //Insert a character at some Index
        sb.insert(2, 'o');
        System.out.println(sb);

        //Append means to add something the end
        sb.append("Devel");
        System.out.println(sb);

        /*Get Char
        sb.setCharAt(2, 'm'); */

        sb.delete(0,1);  // delete char
        System.out.println(sb);
        System.out.println(sb.length());

        }
}
```

Output:

```
C:\Users\dhote\OneDrive\Documents>java stringlength
joony
joonyDevel
oonyDevel
9
```

B) StringTokenizer

Program:

```
import java.util.StringTokenizer;
public class StringTokenizer1
{
 /* Driver Code */
 public static void main(String args[])
 {
   /* StringTokenizer object */
   StringTokenizer st = new StringTokenizer("I am from MCA Depertment"," ");
     /* Checks if the String has any more tokens */
     while (st.hasMoreTokens())
     {
         System.out.println(st.nextToken());
     }
 }
}
```

Output:

```
C:\Users\dhote\OneDrive\Documents>java StringTokenizer1
I
am
from
MCA
Depertment
```

PIRENS Institute of Business Management and Administration, Loni BK.		
Roll Number:	**Date: / /**	**Sign:**
Student Name:		
Subject Name: IT11 JAVA PROGRAMMING		
Program Title: Q.5 Test any five of standard exception and user Defined Custom Exceptions in java		

Program:

```
import java.io.*;
class CheckedExceptionExample {
   public static void main(String args[]){
      FileInputStream file_data = null;
      file_data = new FileInputStream("C:/Users/dhote/OneDrive/Desktop/Hello.txt");
      int m;
      while(( m = file_data.read() ) != -1)
           {
         System.out.print((char)m);
      }
      file_data.close();
   }
}
```

Output:

```
C:\Users\dhote\OneDrive\Documents>javac CheckedExceptionExample.java
CheckedExceptionExample.java:5: error: unreported exception FileNotFoundException; must be caught or declared to be thro
wn
        file_data = new FileInputStream("C:/Users/dhote/OneDrive/Desktop/Hello.txt");
                    ^
CheckedExceptionExample.java:7: error: unreported exception IOException; must be caught or declared to be thrown
        while(( m = file_data.read() ) != -1)
                                ^
CheckedExceptionExample.java:11: error: unreported exception IOException; must be caught or declared to be thrown
        file_data.close();
                  ^
3 errors
```

B) Custom Excetpion:

Program:

```
import java.io.*;
class CheckedExceptionExample {
    public static void main(String args[]) throws IOException{
        FileInputStream file_data = null;
        file_data = new FileInputStream("C:/Users/dhote/OneDrive/Desktop/Hello.txt");
        int m;
        while(( m = file_data.read() ) != -1)
             {
            System.out.print((char)m);
        }
        file_data.close();
    }
}
```

Output:

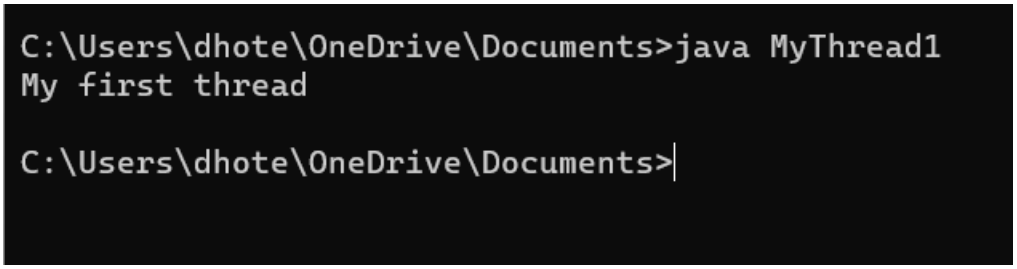
```
C:\Users\dhote\OneDrive\Documents>java CheckedExceptionExample.java
Strong
C:\Users\dhote\OneDrive\Documents>
```

PIRENS Institute of Business Management and Administration, Loni BK.
Roll Number: **Date: / /** **Sign:**
Student Name:
Subject Name: IT11 JAVA PROGRAMMING
Program Title: Q.6 Thread creation and design application by using extending the thread class implementing the runnable interface. Application of multithreading in java

Program:

```
public class MyThread1
{
// Main method
public static void main(String argvs[])
	{
		// creating an object of the Thread class using the constructor Thread(String name)
		Thread t= new Thread("My first thread");

		// the start() method moves the thread to the active state
		t.start();
		// getting the thread name by invoking the getName() method
		String str = t.getName();
		System.out.println(str);
	}
}
```

Output:



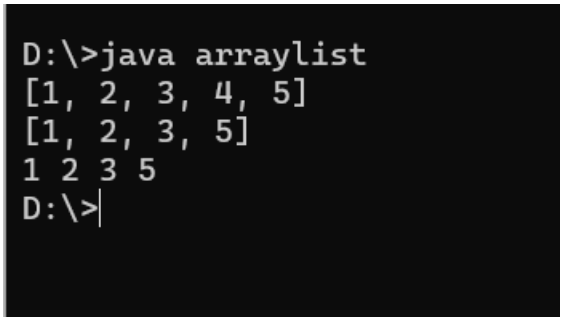
PIRENS Institute of Business Management and Administration, Loni BK.		
Roll Number:	**Date: / /**	**Sign:**
Student Name:		
Subject Name: IT11 JAVA PROGRAMMING		
Program Title: Q.7 Design java application using Collection in java such as Array List, Link List.		

Program:-

Array List:

```
import java.io.*;
import java.util.*;

class arraylist
{
	public static void main(String[] args)
	{
		ArrayList<Integer> al = new ArrayList<Integer>();


		for (int i = 1; i <= 5; i++)
			al.add(i);

		System.out.println(al);

		// Remove element at index 3
		al.remove(3);

		// Displaying the ArrayList
		// after deletion
		System.out.println(al);

		// Printing elements one by one
		for (int i = 0; i < al.size(); i++)
			System.out.print(al.get(i) + " ");
	}
}
```

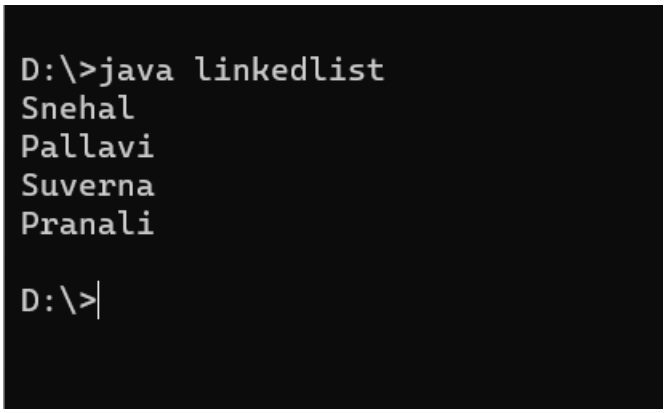
Output:



Linked List ;

```
import java.io.*;
import java.util.*;

import java.util.*;
public class linkedlist
{
public static void main(String args[])
	{
		LinkedList<String> al=new LinkedList<String>();
		al.add("Snehal");
		al.add("Pallavi");
		al.add("Suverna");
		al.add("Pranali");
		Iterator<String> itr=al.iterator();
		while(itr.hasNext())
	{
		System.out.println(itr.next());
	}
	}
}
```

Output:

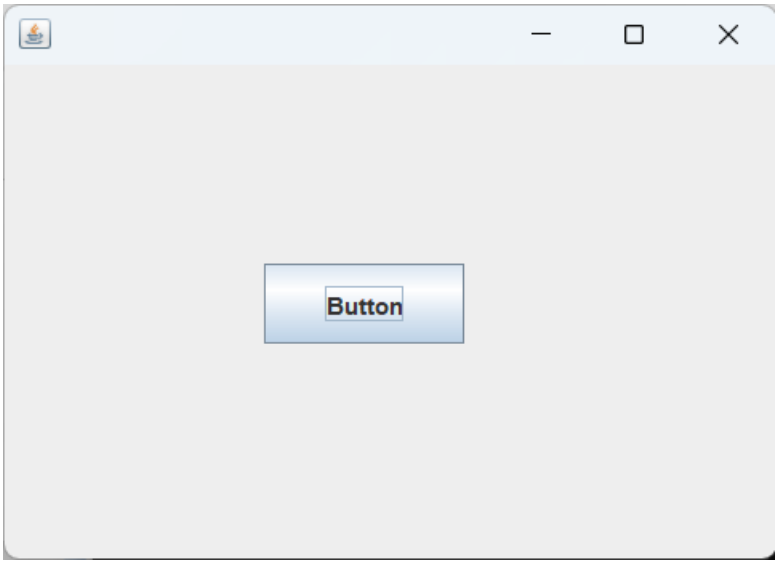


PIRENS Institute of Business Management and Administration, Loni BK.		
Roll Number:	**Date: / /**	**Sign:**
Student Name:		
Subject Name: IT11 JAVA PROGRAMMING		
Program Title: Q.8 Design GUI based java application using AWT, Swing with Event Handling		

Program:

```
import javax.swing.*;
public class swingeg
{
	public static void main(String[] args)
	{
		JFrame f=new JFrame();//creating instance of JFrame

		JButton b=new JButton("Button");//creating instance of JButton
		b.setBounds(130,100,100, 40);//x axis, y axis, width, height

		f.add(b);//adding button in JFrame

		f.setSize(400,500);//400 width and 500 height
		f.setLayout(null);//using no layout managers
		f.setVisible(true);//making the frame visible
	}
}
```

Output:



PIRENS Institute of Business Management and Administration, Loni BK.		
Roll Number:	**Date: / /**	**Sign:**
Student Name:		
Subject Name: IT11 JAVA PROGRAMMING		
Program Title: Q..9 Design a and implement JDBC applications.		

Program:

```
mport java.sql.*;
public class FirstExample
{
        static final String DB_URL = "jdbc:mysql://localhost/TUTORIALSPOINT";
        static final String USER = "guest";
        static final String PASS = "guest123";
        static final String QUERY = "SELECT id, first, last, age FROM Employees";
 public static void main(String[] args)
{
 // Open a connection

        try{
        System.out.println("Connecting to database… \n");
        Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
        System.out.println("Creating Statement… \n");
        Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery(QUERY);) {
        // Extract data from result set
        while (rs.next()) {
        // Retrieve by column name
        System.out.print("ID: " + rs.getInt("id"));
        System.out.print(", Age: " + rs.getInt("age"));
        System.out.print(", First: " + rs.getString("first"));
        System.out.println(", Last: " + rs.getString("last"));
         }
        }
        catch (SQLException e)
        {
                e.printStackTrace();
        }
}
}
```

Output:

```
Connecting to database...
Creating statement...
ID: 100, Age: 18, First: Zara, Last: Ali
ID: 101, Age: 25, First: Mahnaz, Last: Fatma
ID: 102, Age: 30, First: Zaid, Last: Khan
ID: 103, Age: 28, First: Sumit, Last: Mittal
```

PIRENS Institute of Business Management and Administration, Loni BK.		
Roll Number:	**Date: / /**	**Sign:**
Student Name:		
Subject Name: IT11 JAVA PROGRAMMING		
Program Title: Q..10 Design and implement servlet applications.		

Program:

```
// importing the javax.servlet package
// importing java.io package for PrintWriter
import javax.servlet.*;
import java.io.*;
// now creating a servlet by implementing Servlet interface
public class LifeCycleServlet implements Servlet
{
        ServletConfig config = null;
        // init method
        public void init(ServletConfig sc)
                {
                        config = sc;
                        System.out.println("in init");
                }
 // service method
 public void service(ServletRequest req, ServletResponse res)
 throws ServletException, IOException
 {
        res.setContenttype("text/html");
        PrintWriter pw = res.getWriter();
        pw.println("<h2>hello from life cycle servlet</h2>");
        System.out.println("in service");
 }
 // destroy method
 public void destroy()
        {
                System.out.println("in destroy");
        }
public String getServletInfo()
         {
                return "LifeCycleServlet";
        }
 public ServletConfig getServletConfig()
        {
                return config; // getServletConfig
        }
}
```

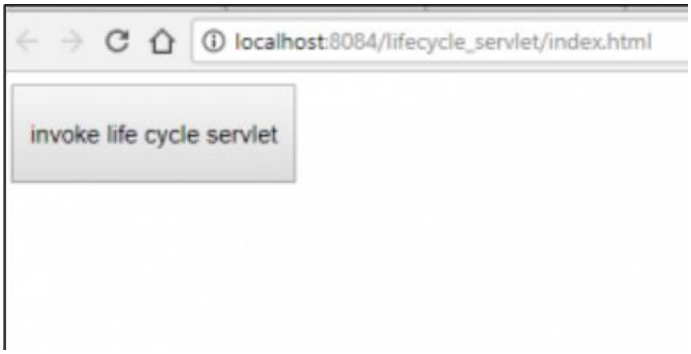
HTML code :

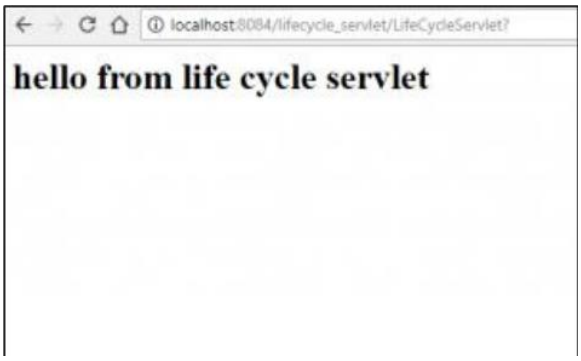
```
<html>
<body>
 <form action="LifeCycleServlet">
 <input type="submit" value="invoke life cycle servlet">
 </form>
</body>
</html>
```

web.xml file code :

```
<?xml version="1.0" encoding="UTF=8"?>
<web-app>
 <servlet>
 <servlet-name>LifeCycleServlet</servlet-name>
 <servlet-class>LifeCycleServlet</servlet-class>
 </servlet>
 <servlet-mapping>
 <servlet-name>LifeCycleServlet</servlet-name>
 <url-pattern>/LifeCycleServlet</url-pattern>
 </servlet-mapping>
 <session-config>
 <session-timeout>
 30
 </session-config>
</web-app>
```

Output:





PIRENS Institute of Business Management and Administration, Loni BK.		
Roll Number:	**Date: / /**	**Sign:**
Student Name:		
Subject Name: IT11 JAVA PROGRAMMING		
Program Title: Q.11 Design and implement JSP application		

Program:

```
<HEAD>
<TITLE>JSP Example</TITLE>
</HEAD>
<BODY BGCOLOR="ffffcc">
<CENTER>
<H2>Date and Time</H2>
<% java.util.Date today = new java.util.Date();
out.println("Today's date is: "+today);%>
</CENTER>
</BODY>
</HTML>
```

Output: