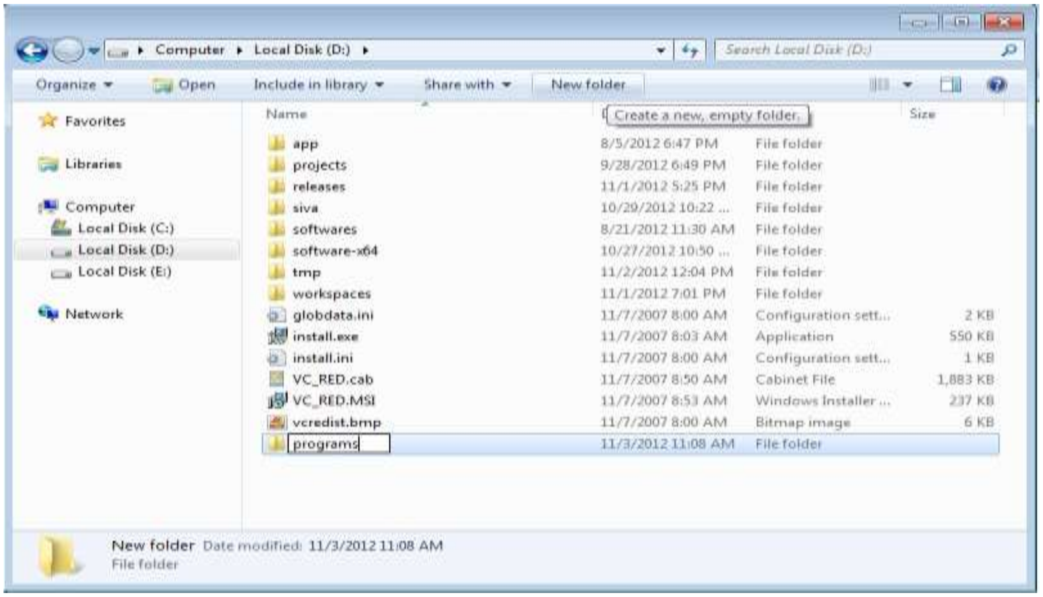
STEPS FOR COMPILING AND RUNNING A JAVA PROGRAM

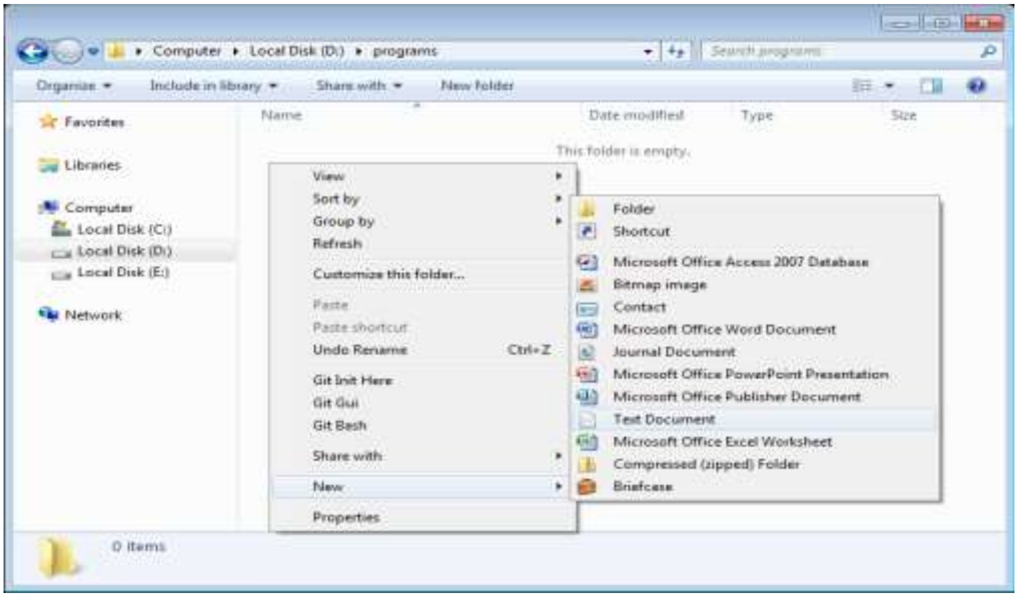
1. Create a directory: Open the windows explorer and go to the D drive. Inside that create a directory called programs. This will be the directory you will use for storing all your java programs. This step you have to do only once per computer, it is not necessary to create a new directory for every Java program.



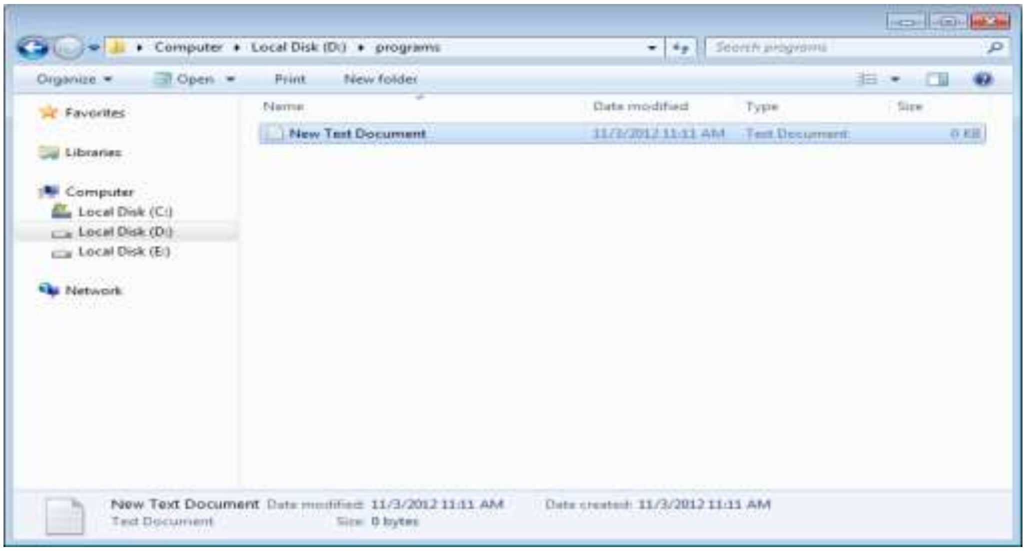
Name			Size
app	8/5/2012 6:47 PM	File folder	
projects	9/28/2012 6:49 PM	File folder	
releases	11/1/2012 5:25 PM	File folder	
siva	10/29/2012 10:22 ...	File folder	
softwares	8/21/2012 11:30 AM	File folder	
software-x64	10/27/2012 10:50 ...	File folder	
tmp	11/2/2012 12:04 PM	File folder	
workspaces	11/1/2012 7:01 PM	File folder	
globdata.ini	11/7/2007 8:00 AM	Configuration sett...	2 KB
install.exe	11/7/2007 8:03 AM	Application	550 KB
install.ini	11/7/2007 8:00 AM	Configuration sett...	1 KB
VC_RED.cab	11/7/2007 8:50 AM	Cabinet File	1,883 KB
VC_RED.MSI	11/7/2007 8:53 AM	Windows Installer ...	237 KB
vcredist.bmp	11/7/2007 8:00 AM	Bitmap image	6 KB
programs	11/3/2012 11:08 AM	File folder	

New folder Date modified: 11/3/2012 11:08 AM
File folder

2. Create a new text document: Go to the programs directory in windows explorer and create a new text file by right clicking in the empty area and use the option 'New' -> 'Text Document'.

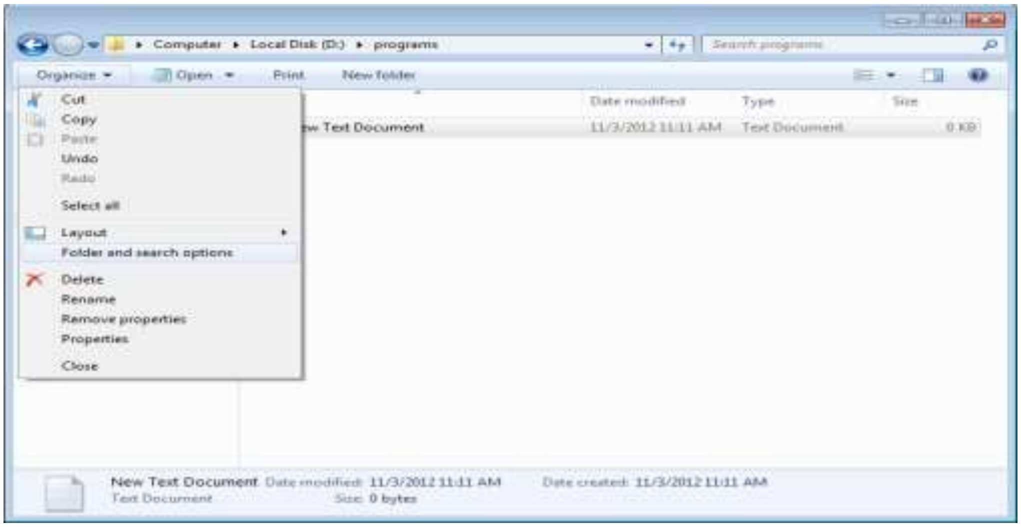


3. Change the folder options (if required): If the newly created file is 'New Text Document' instead of 'New Text Document.txt', then you need to this additional step of changing the folder options to show the file extension. Please go to the menu option 'Organize' -> 'Folder And Search Options', it launches a dialog with name 'Folder Options'. Go to the tab 'View' and un-select the option 'Hide extensions for known file types'. Click 'OK' on the dialog. Then it should show the file name as 'New Text Document.txt'.



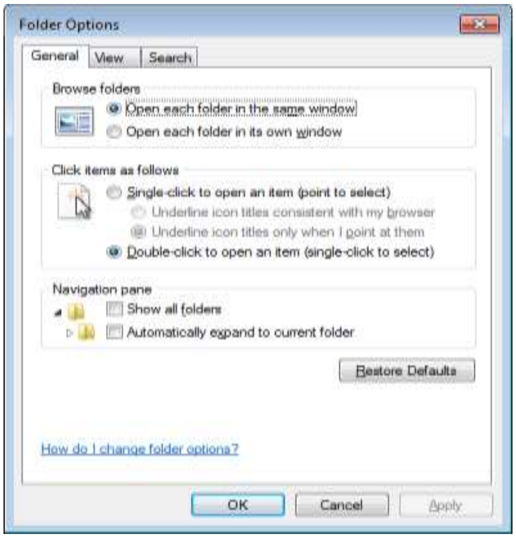
Name	Date modified	Type	Size
New Text Document	11/3/2012 11:11 AM	Text Document	0 KB

New Text Document Date modified: 11/3/2012 11:11 AM Date created: 11/3/2012 11:11 AM
Text Document Size: 0 bytes



Name	Date modified	Type	Size
New Text Document	11/3/2012 11:11 AM	Text Document	0 KB

New Text Document Date modified: 11/3/2012 11:11 AM Date created: 11/3/2012 11:11 AM
Text Document Size: 0 bytes



Browse folders

● Open each folder in the same window

○ Open each folder in its own window

Click items as follows

○ Single-click to open an item (point to select)

○ Underline icon titles consistent with my browser

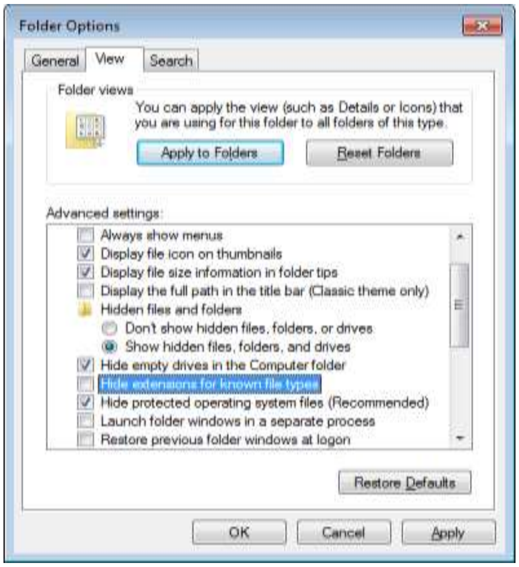
● Underline icon titles only when I point at them

● Double-click to open an item (single-click to select)

Navigation pane

□ Show all folders

□ Automatically expand to current folder



Folder views

You can apply the view (such as Details or Icons) that you are using for this folder to all folders of this type.

Advanced settings:

□ Always show menus

☑ Display file icon on thumbnails

☑ Display file size information in folder tips

□ Display the full path in the title bar (Classic theme only)

Hidden files and folders

○ Don't show hidden files, folders, or drives

● Show hidden files, folders, and drives

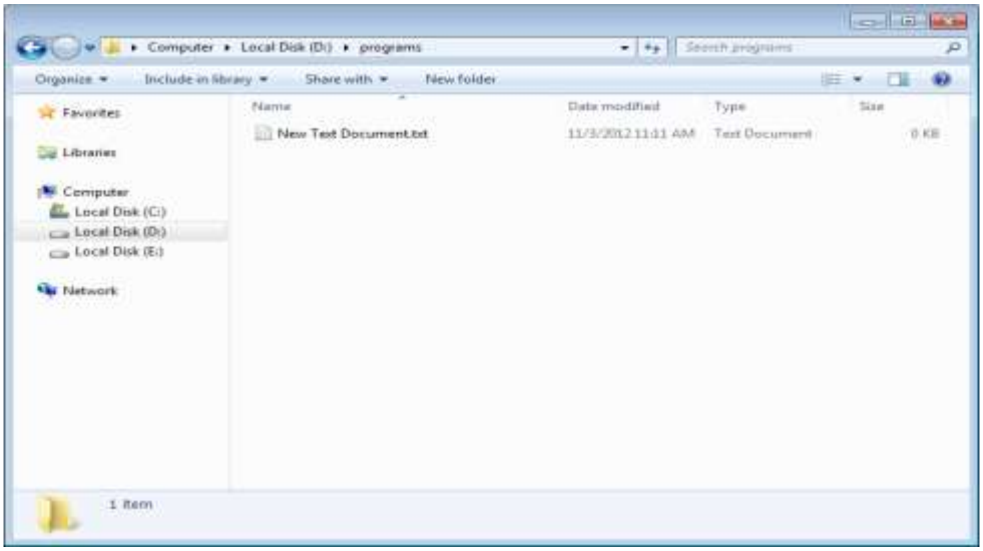
☑ Hide empty drives in the Computer folder

□ Hide extensions for known file types

☑ Hide protected operating system files (Recommended)

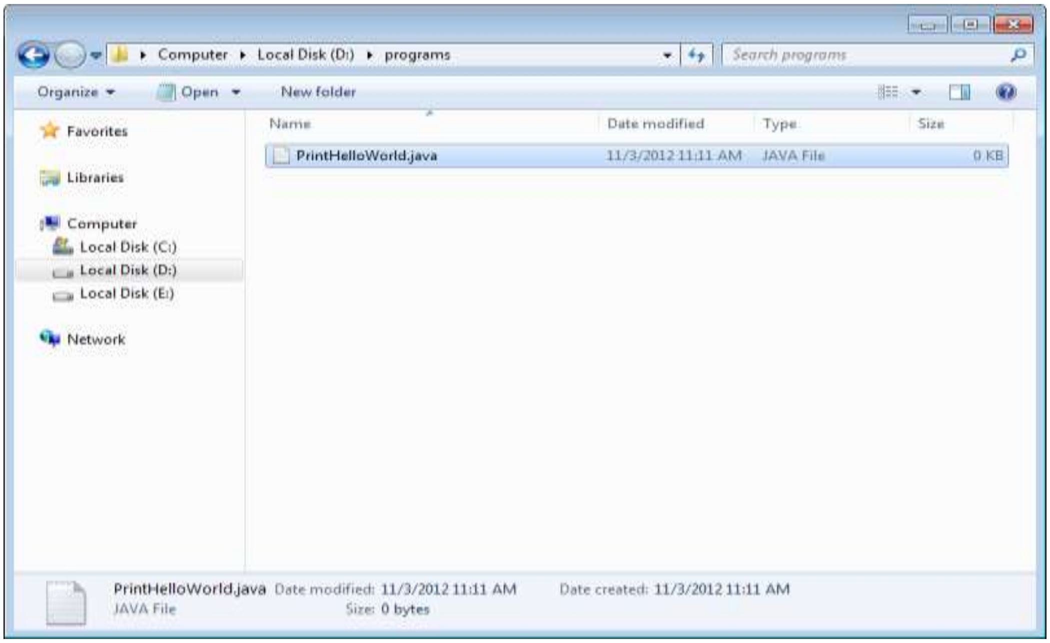
□ Launch folder windows in a separate process

□ Restore previous folder windows at logon



Name	Date modified	Type	Size
New Text Document.txt	11/3/2012 11:11 AM	Text Document	0 KB

4. Rename the text document to program name: Right click on 'New Text Document.txt' and rename it to 'PrintHelloWorld.java'. You should see the .java extension after you rename it. The type should be 'JAVA file' instead 'Text Document'.

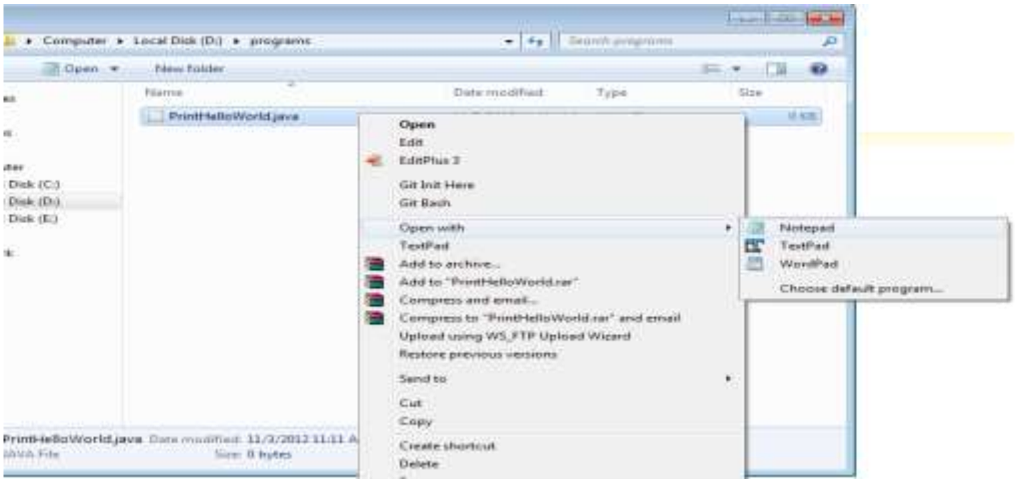


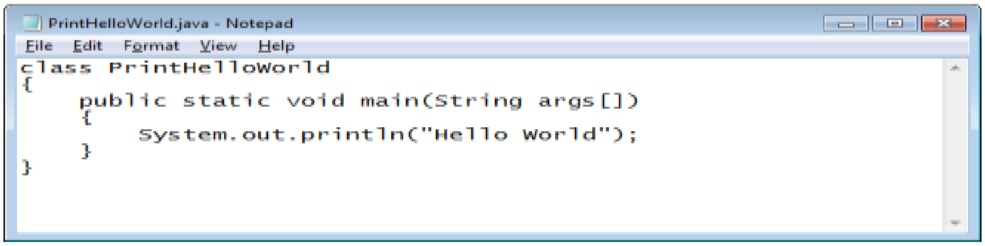
Name	Date modified	Type	Size
PrintHelloWorld.java	11/3/2012 11:11 AM	JAVA File	0 KB

PrintHelloWorld.java Date modified: 11/3/2012 11:11 AM Date created: 11/3/2012 11:11 AM
JAVA File Size: 0 bytes

5. Why we called it 'PrintHelloWorld.java': Since in this example we will be copying the code from *Java Sample Program - Simple Hello World Program in Java* and the program name their is PrintHelloWorld. Please note that the file name PrintHelloWorld.java and the program name PrintHelloWorld should be exactly same and they are case sensitive. Having the file names like 'printhelloworld.java' or 'Printhelloworld.java' or 'printworld.java' is not allowed. The spelling should be exactly same and the case of letters (CAPITAL or small) should also be same.

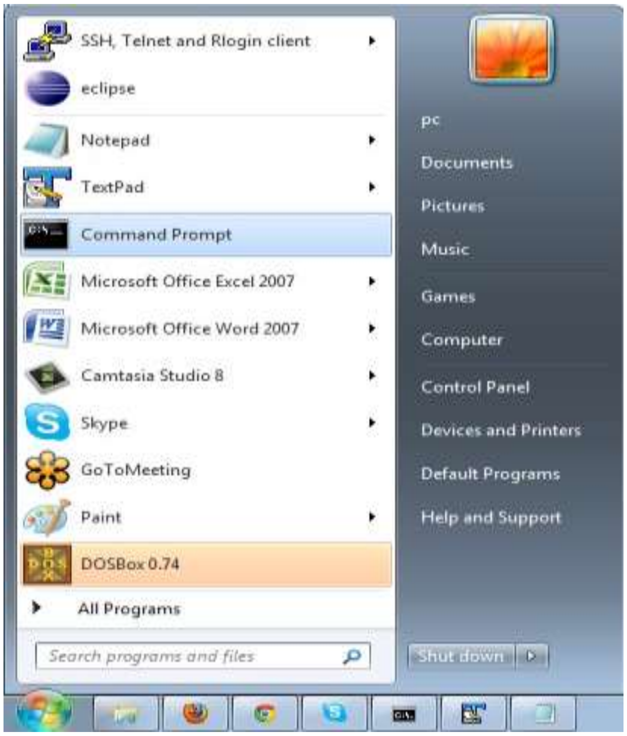
6. Copy the Java program into the file 'PrintHelloWorld.java': Open the file 'PrintHelloWorld.java' using 'Notepad' and copy the program code from *Java Sample Program - Simple Hello World Program In Java* and save it using 'File'->'Save'.



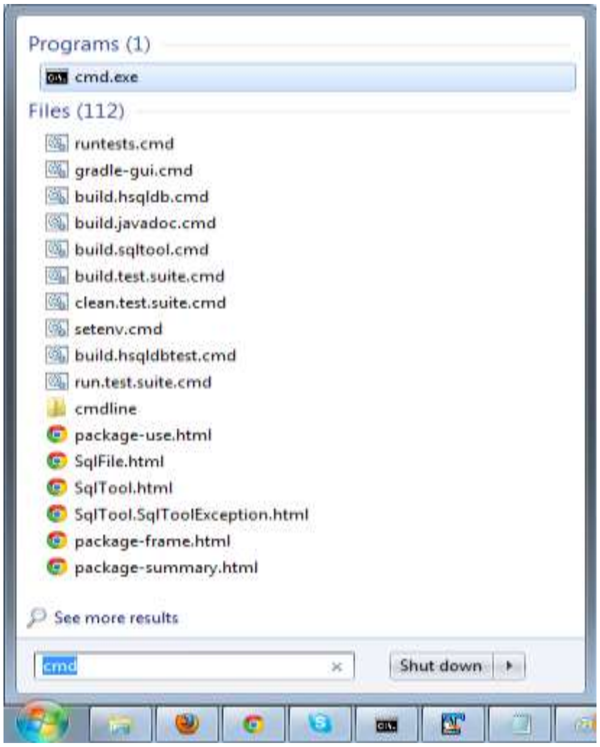


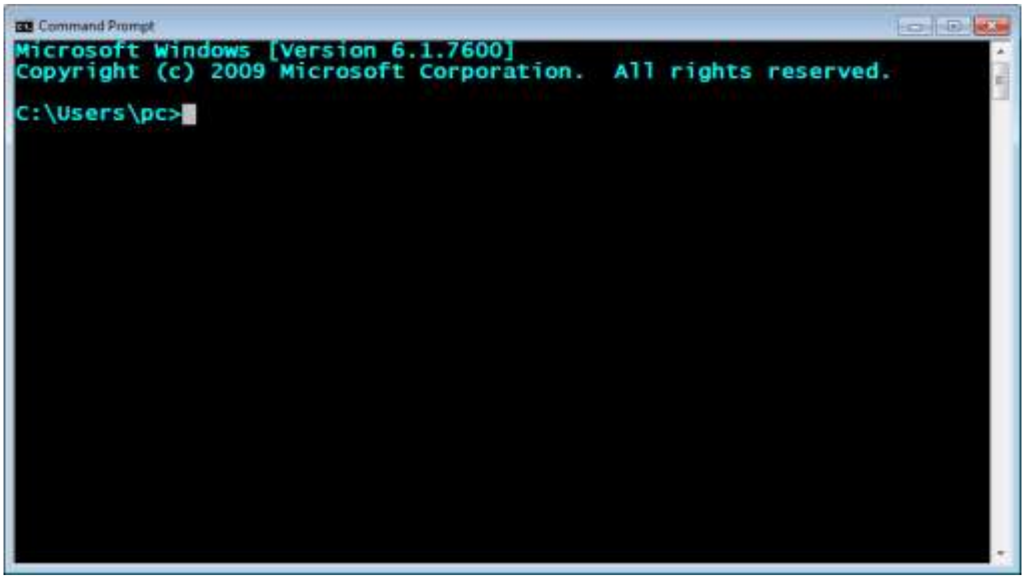
```
class PrintHelloWorld
{
    public static void main(String args[])
    {
        System.out.println("Hello World");
    }
}
```

7. Launch the command prompt: Go to the 'Start' menu and launch the 'Command Prompt' program. Or run the 'cmd' program to launch the command prompt.



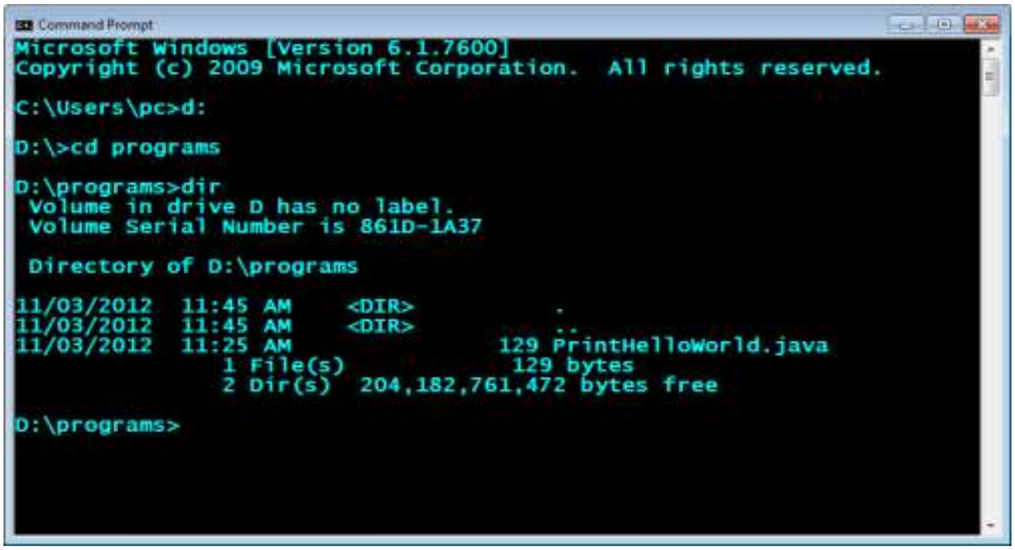
or





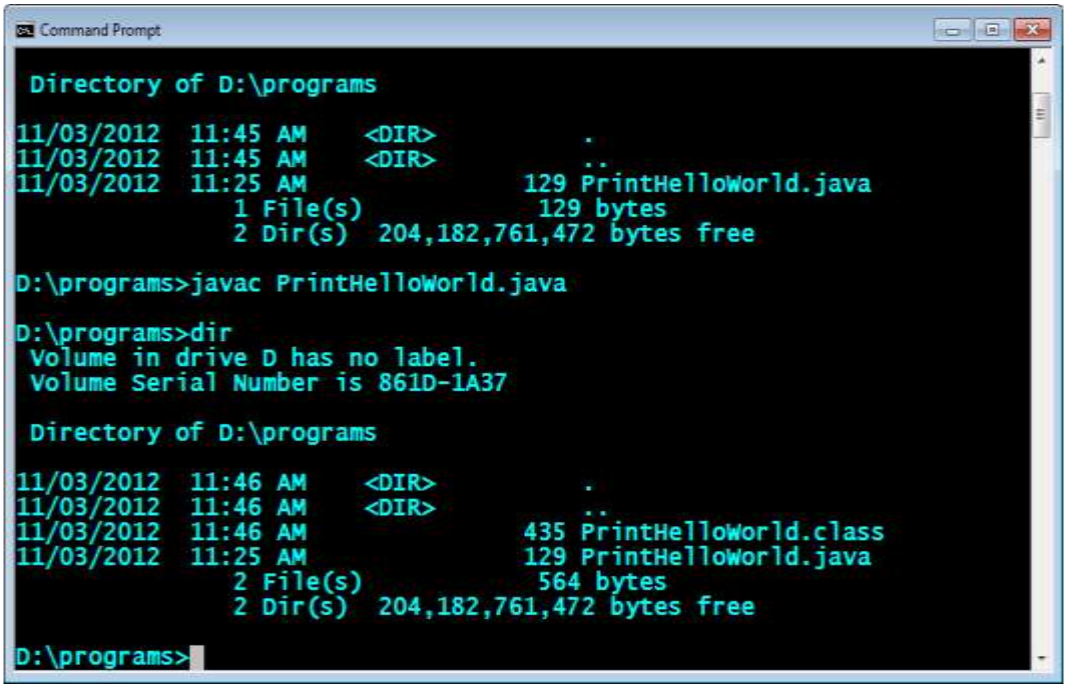
```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\pc>
```

8. Go to the required directory: Use the command 'd:' to go the required drive, 'cd programs' to go to the programs directory, 'dir' to see the list of files. The directory listing should show the file 'PrintHelloWorld.java'



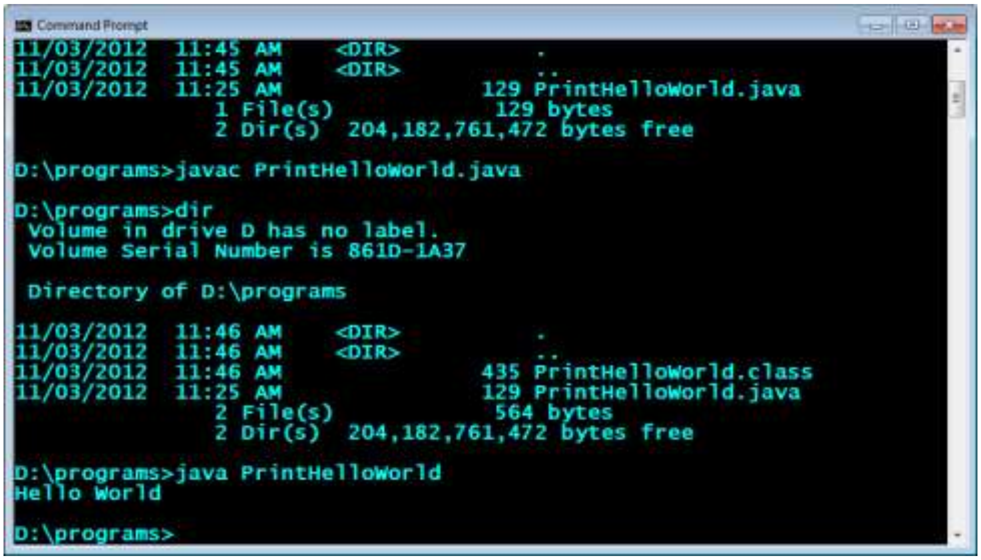
```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\pc>d:

D:\>cd programs

D:\programs>dir
 Volume in drive D has no label.
 Volume Serial Number is 861D-1A37

 Directory of D:\programs

11/03/2012  11:45 AM    <DIR>          .
11/03/2012  11:45 AM    <DIR>          ..
11/03/2012  11:25 AM               129 PrintHelloWorld.java
               1 File(s)            129 bytes
               2 Dir(s)  204,182,761,472 bytes free

D:\programs>
```

9. Compiling the Java program: On the command prompt use the command 'javac PrintHelloWorld.java' to compile the program. It should compile without any errors. If it says 'not a recognized program', then it means the java is not installed or it is not proper. Go to *Installation of Java on your PC* for installing the same. If there are no errors, then run the command 'dir' and see that a new file 'PrintHelloWorld.class' is created. This .class file is created when we compiled the program and it contains the bytecode. You should also see this file in the file explorer.



```
 Directory of D:\programs

11/03/2012  11:45 AM    <DIR>          .
11/03/2012  11:45 AM    <DIR>          ..
11/03/2012  11:25 AM               129 PrintHelloWorld.java
               1 File(s)            129 bytes
               2 Dir(s)  204,182,761,472 bytes free

D:\programs>javac PrintHelloWorld.java

D:\programs>dir
 Volume in drive D has no label.
 Volume Serial Number is 861D-1A37

 Directory of D:\programs

11/03/2012  11:46 AM    <DIR>          .
11/03/2012  11:46 AM    <DIR>          ..
11/03/2012  11:46 AM               435 PrintHelloWorld.class
11/03/2012  11:25 AM               129 PrintHelloWorld.java
               2 File(s)            564 bytes
               2 Dir(s)  204,182,761,472 bytes free

D:\programs>
```

10. Run the Java program: On the command prompt use the command 'java PrintHelloWorld' to run the program. It should print the output 'Hello World' on the screen. This means we are able to successfully compile and run a simple java program.



```
11/03/2012  11:45 AM    <DIR>          .
11/03/2012  11:45 AM    <DIR>          ..
11/03/2012  11:25 AM               129 PrintHelloWorld.java
               1 File(s)            129 bytes
               2 Dir(s)  204,182,761,472 bytes free

D:\programs>javac PrintHelloWorld.java

D:\programs>dir
 Volume in drive D has no label.
 Volume Serial Number is 861D-1A37

 Directory of D:\programs

11/03/2012  11:46 AM    <DIR>          .
11/03/2012  11:46 AM    <DIR>          ..
11/03/2012  11:46 AM               435 PrintHelloWorld.class
11/03/2012  11:25 AM               129 PrintHelloWorld.java
               2 File(s)            564 bytes
               2 Dir(s)  204,182,761,472 bytes free

D:\programs>java PrintHelloWorld
Hello World

D:\programs>
```

MAKING FURTHER CHANGES TO THE PRINTHELLOWORLD PROGRAM

To make further changes to the program:

- Open 'PrintHelloWorld.java' in notepad, make the necessary changes, save it using 'File'->'Save'
- Compile in command prompt using 'javac PrintHelloWorld.java'
- Run using the command 'java PrintHelloWorld'

CREATING A NEW PROGRAM CALLED SayHelloHi

- Create a new text document using 'New' -> 'Text Document' in file explorer, rename it to 'SayHelloHi.java'
- Write code in the file 'SayHelloHi.java'. The class name given in the file should be 'SayHelloHi' which is same as the program name.
- Save the file using 'File'->'Save'.
- Compile in command prompt using 'javac SayHelloHi.java'
- Run using 'java SayHelloHi'.

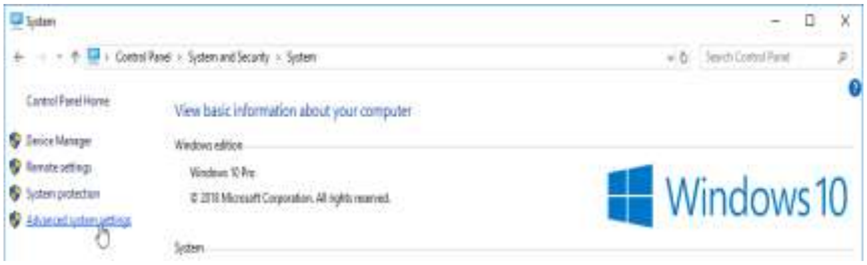
SETTING UP THE ENVIRONMENT FOR FUTURE PROGRAMMING SESSIONS

- Open the directory D:\programs in windows (or file) explorer. This is where all your programs are present.
- Open the command prompt using 'Start'->'Command Prompt' or by running 'cmd'. Go to your programs directory using the commands 'd:', 'cd programs'. Use 'dir' to see the list of all programs and the .class files created.
- Run the command 'javac PrintHelloWorld.java' and 'java PrintHelloWorld' to confirm that your java installation is still proper.

STEPS TO INSTALL JAVA DEVELOPMENT KIT IN WINDOWS

1. Go to "System Properties" (Can be found on Control Panel > System and Security > System -> Advanced System Settings)
2. Click on the "Environment variables" button under the "Advanced" tab
3. Then, select the "Path" variable in System variables and click on the "Edit" button
4. Click on the "New" button and add the path where Java is installed, followed by **\bin**. By default, Java is installed in C:\Program Files\Java\jdk-11.0.1 (If nothing else was specified when you installed it). In that case, You will have to add a new path with: **C:\ProgramFiles\Java\jdk-11.0.1\bin** Then, click "OK", and save the settings
5. At last, open Command Prompt (cmd.exe) and type **java -version** to see if Java is running on your machine

STEP 1



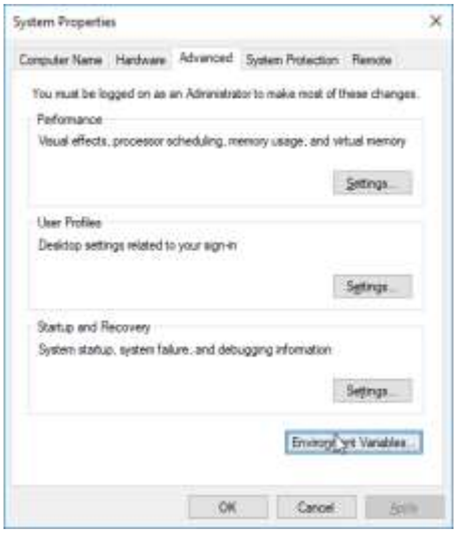
View basic information about your computer

Windows edition

Windows 10 Pro

© 2018 Microsoft Corporation. All rights reserved.

STEP 2



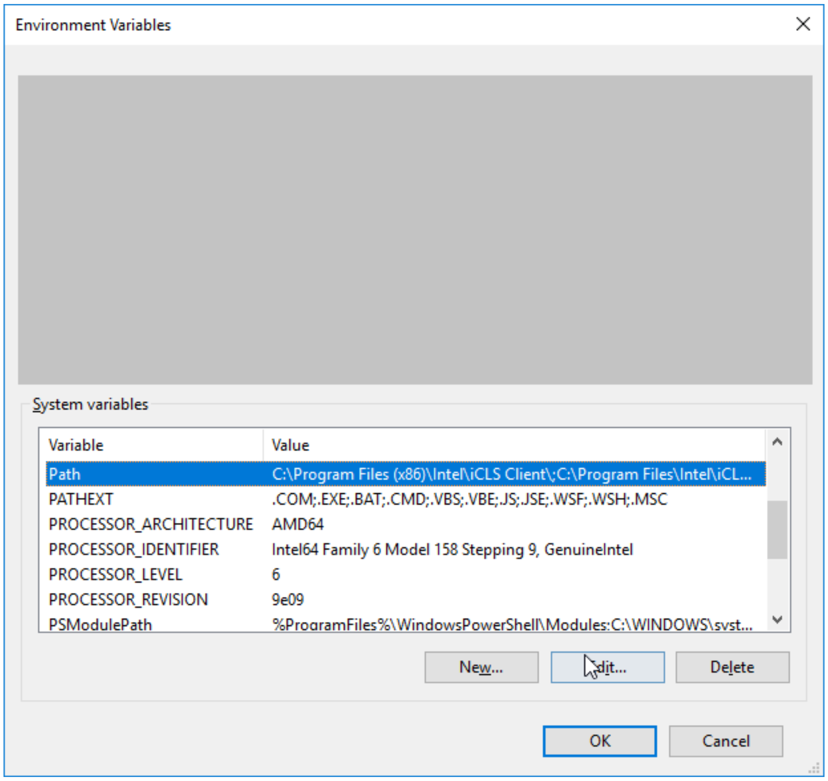
You must be logged on as an Administrator to make most of these changes.

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

User Profiles: Desktop settings related to your sign-in

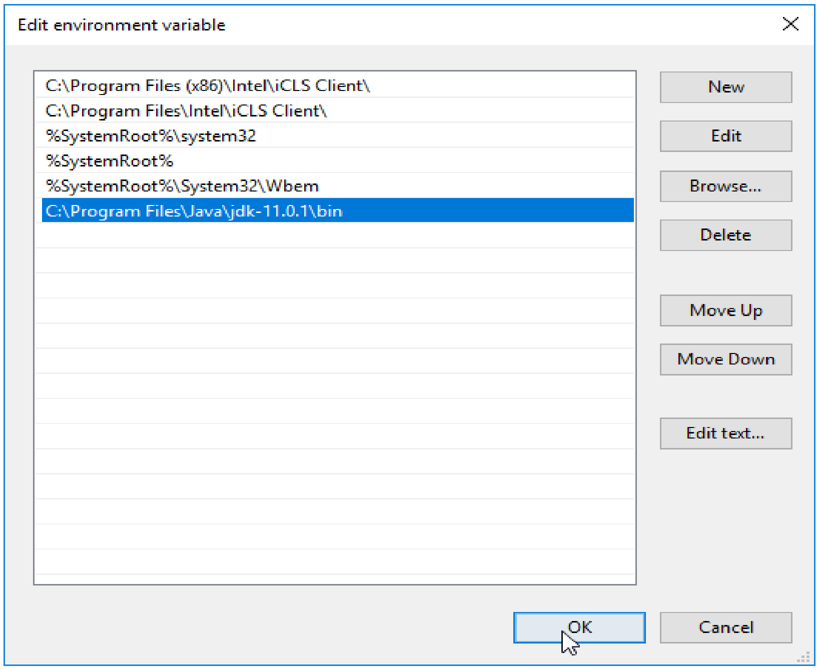
Startup and Recovery: System startup, system failure, and debugging information

STEP 3



Variable	Value
Path	C:\Program Files (x86)\Intel\iCLS Client\;C:\Program Files\Intel\iCL...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 158 Stepping 9, GenuineIntel
PROCESSOR_LEVEL	6
PROCESSOR_REVISION	9e09
PSModulePath	%ProgramFiles%\WindowsPowerShell\Modules;C:\WINDOWS\syst...

STEP 4



```
C:\Program Files (x86)\Intel\iCLS Client\
C:\Program Files\Intel\iCLS Client\
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
C:\Program Files\Java\jdk-11.0.1\bin
```

STEP 5

Write the following in the command line (cmd.exe):

```
C:\Users\Your Name>java -version
```

If Java was successfully installed, you will see something like this (depending on version):

```
java version "11.0.1" 2018-10-16 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.1+13-LTS)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.1+13-LTS, mixed mode)
```

Ex.No:01	**SEARCHING AND SORTING ALGORITHMS**
Date:	

AIM:

To develop a Java Program to perform searching and sorting.

ALGORITHM:

Step1: Start the program.

Step 2: Declare class with members and methods

Step 3: Initialize an array of elements.

Step 4: Perform searching and sorting by iterating over the array

Step 5: Stop the program

PROGRAM:

Sequential Search:

```
Sequential.java import java.io.*;
import java.util.*;
public class Sequential
{
public static void main(String args[])
{
int[] exampleset = {2, 9, 6, 7, 4, 5, 3, 0, 1, 10};
System.out.println("Enter Your target integer in the array :");
Scanner scn = new Scanner(System.in);
int target = scn.nextInt();
 sequentialSearch(exampleset, target);
}
public static void sequentialSearch(int[] ref1, int ref2)
{
int index = -1;
for (int i = 0; i < ref1.length; i++)
{
if (ref1[i] == ref2)
{
index = i; break;
}
}
if (index == -1)
{
System.out.println("Your target integer does not exist in the array");
}
else
{
System.out.println("Your target integer is in index " + index + " of the array");
}
}
}
```

Output:

```
Enter Your target integer in the array: 5 Your target integer is in index 5 of the array
```

Binary Search :

```
import java.io.*;

 import java.util.*;

public class Binary

{
public static void main(String args[])
{
int[] exampleset = {1, 11, 24, 34, 67, 89, 102};

System.out.println("Enter Your target integer in the array :");

Scanner scn = new Scanner(System.in);

int target = scn.nextInt();

 binarySearch(exampleset, target);

}
public static void binarySearch(int[] ref1, int ref2)
{
int index = -1;

int lowEnd = 0;

int highEnd = ref1.length - 1;

while (highEnd >= lowEnd)

{
int middle = (lowEnd + highEnd) / 2;

if (ref1[middle] == ref2)

{
index = middle; break;

}
else if (ref1[middle] < ref2)
{
lowEnd = middle + 1;
}
else if (ref1[middle] > ref2)
{
highEnd = middle - 1;
}
}
if (index == -1)
{
System.out.println("Your target integer does not exist in the array");
}
else
{
System.out.println("Your target integer is in index " + index + " of the array");
}
}
}
```

Output:

```
Enter Your target integer in the array :89 Your target integer is in index 5 of the array
```

Selection Sort :

```
import java.io.*;
public  class SelectionSort
{
public static void selectionSort(int[] arr)
{
for (int i = 0; i < arr.length - 1; i++)
{
int index = i;
for (int j = i + 1; j < arr.length; j++)
{
if (arr[j] < arr[index])
{
index = j; //searching for lowest index
}
}
int smallerNumber = arr[index]; arr[index] = arr[i];

arr[i] = smallerNumber;
}
}
public static void main(String a[])
{
int[] arr1 = {9, 14, 3, 2, 43, 11, 58, 22};
System.out.println("Before Selection Sort"); for (int i: arr1)

{
System.out.print(i + " ");
}
System.out.println();
selectionSort(arr1);
System.out.println("After Selection Sort");
for (int i: arr1)
{
System.out.print(i + " ");
}
}
}
```

Output:

```
Before Selection Sort 9 14 3 2 43 11 58 22
After Selection Sort 2 3 9 11 14 22 43 58
```

Insertion Sort :

```
import java.io.*;
public class InsertionSort
{
public static void insertionSort(int array[])
{
int n = array.length;
for (int j = 1; j < n; j++)
{
int key = array[j];
int i = j - 1;
while ((i > -1) && (array[i] > key))
{
array[i + 1] = array[i]; i--;
}
array[i + 1] = key;
}
}
public static void main(String a[])
{
int[] arr1 = {9, 14, 3, 2, 43, 11, 58, 22};
System.out.println("Before Insertion Sort");
for (int i: arr1)
{
System.out.print(i + " ");
}
System.out.println();
insertionSort(arr1);
System.out.println("After Insertion Sort");
for (int i: arr1)
{
System.out.print(i + " ");
}
}
}
```

Output:

```
Before Insertion Sort 9 14 3 2 43 11 58 22

After Insertion Sort 2 3 9 11 14 22 43 58
```

RESULT:

Thus the java program to perform searching and sorting was executed and the output is obtained.

Ex.No:02	**STACK AND QUEUE DATA STRUCTURES USING CLASSES AND OBJECTS**
Date:	

AIM:

To develop a java application to implement stack and queue data structures using class and objects.

ALGORITHM:

Step 1: Start the program.

Step 2: Create class to implement stack and queue data structure.

Step 3: Perform push and Pop operations in stack.

Step 4: Perform enqueue and dequeue operations in queue.

Step 5: Create object to invoke the methods in class.

Step 6: Stop the program

PROGRAM:

Stack :

```
import java.util.EmptyStackException;
import java.util.Stack;
public class StackPushPopExample
{
public static void main(String args[])
{
Stack < Integer > stk = new Stack < > ();
System.out.println("stack: " + stk);
pushelement(stk, 20);
pushelement(stk, 13);
pushelement(stk, 89);
pushelement(stk, 90);
pushelement(stk, 11);
pushelement(stk, 45);
pushelement(stk, 18);
popelement(stk);
 popelement(stk);
try
{
popelement(stk);
}
catch (EmptyStackException e)
{
System.out.println("empty  stack");
}
}
static void pushelement(Stack stk, int x)
{
stk.push(new Integer(x));
System.out.println("push -> " + x);
System.out.println("stack: " + stk);
}
static void popelement(Stack stk)
{
System.out.print("pop -> ");
Integer x = (Integer) stk.pop();
System.out.println(x);
System.out.println("stack: " + stk);
}
}
```

Output:

```
stack: [] push -> 20

stack: [20]
push -> 13
stack: [20, 13]
push -> 89
stack: [20, 13, 89]
push -> 90
stack: [20, 13, 89, 90]
push -> 11
stack: [20, 13, 89, 90, 11]
push -> 45
stack: [20, 13, 89, 90, 11, 45]
push -> 18
stack: [20, 13, 89, 90, 11, 45, 18]
pop -> 18
stack: [20, 13, 89, 90, 11, 45]
pop -> 45
stack: [20, 13, 89, 90, 11]
pop -> 11
stack: [20, 13, 89, 90]
```

Queue:

```
import java.util.*;
public class QueueEx1
{
int queueLength = 3;
int items[] = new int[queueLength];
int front = -1;
int back = -1; Boolean isFull()
{
if (back == queueLength - 1)
{
return true;
}
else
{
```

```
return false;
}
}
boolean isEmpty()
{
if (front == -1 && back == -1)
{
return true;
}
else
{
return false;
}
}
void enQueue(int itemValue)
{
if (isFull())
{
System.out.println("Queue is full");
}
else if (front == -1 && back == -1)
{
front = back = 0; items[back] = itemValue;
}
else
{
back++;
items[back] = itemValue;
}
}
void deQueue()
{
if (isEmpty())
{
System.out.println("Queue is empty. Nothing to deQueue");
}
else if (front == back)
{
front = back = -1;
}
else
```

```
{
front++;
}
}
void display()
{
int i;
if (isEmpty())
{
System.out.println("Queue is empty");
}
else
{
for (i = front; i <= back; i++)
{
System.out.print("\t " + items[i]);
}
System.out.print("\n  ");
}
}
void peak()
{
System.out.println("Front value is: " + items[front]);
}
public static void main(String[] args) throws ArrayIndexOutOfBoundsException
{
QueueEx1 myQueue = new QueueEx1();
System.out.println("Enqueue: ");
myQueue.enQueue(3);
myQueue.display();
myQueue.enQueue(2);
myQueue.display();
myQueue.enQueue(1);
myQueue.display();
myQueue.peak();
System.out.println("DeQueue: ");
myQueue.deQueue();
myQueue.display();
myQueue.deQueue();
myQueue.display();
myQueue.peak();
}
}
```

Output:

```
    Enqueue:
        3


        3              2

        3              2        1

Front value is: 3 DeQueue:

2                          1
1

Front value is: 1
```

RESULT:

Thus the java program to perform stack and queue data structures was executed and the Output is obtained.

Ex.No:03	**INHERITANCE IN JAVA**
Date:	

AIM:

To develop a java application with Employee class with Emp_name, Emp_id, Address, Mail_id, Mobile_noasmembers. Inherittheclasses, Programmer, AssistantProfessor, AssociateProfessor and Professor from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 97% of BP as DA, 10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club fund. Generate pay slips for the employees with their gross and net salary.

ALGORITHM:

Step1: Start the program.

Step2: Create the class "Employee" with necessary fields.

Step3: Create a class called programmer that extends employee class.

Step4: Write a method compute pay() to compute employee gross and net salary.

Step 5: Create a classes Asst_Professor, Associate_Professor and Professor that extends employee class.

Step 6: Create a main class PaySlip and create object for all the classes to calculate Grosspay and net pay.

Step7: Stop the program.

PROGRAM:

```
import java.util.Scanner;
class Employee
{
int Emp_id;
String Emp_name; String Address;
String Mail_id; String Mobile_no; Employee()
{
Employee(int id, String name, String addr, String mail, String mob)
{
this.Emp_id = id; this.Emp_name = name;
this.Address = addr;
this.Mail_id = mail;
this.Mobile_no = mob;
}
}
class Programmer extends Employee
{
Double BP, Gross_salary, Net_salary;
public Programmer(int id, String name, String addr, String mail, String mob)
{ super(id, name, addr, mail, mob);
}
void computePay()
{
System.out.println("enter basic pay");
 Scanner input = new Scanner(System.in);
BP = input.nextDouble();
double DA, HRA, PF, Fund;
DA = (BP * 97 / 100);
HRA = (BP * 10 / 100);
 PF = (BP * 12 / 100);
Fund = (BP * 0.1 / 100);
Gross_salary = BP + DA + HRA;
Net_salary = BP + DA + HRA - (PF + Fund);
System.out.println("Emp_Id:" + Emp_id);
System.out.println("Emp_name:" + Emp_name);
System.out.println("Address" + Address);
System.out.println("mail_Id:" + Mail_id);
System.out.println("Mobile_no:" + Mobile_no);
System.out.println("Grosspay:" + Gross_salary);
System.out.println("Netpay:" + Net_salary);
}
}
class Asst_Proffessor extends Employee
{
double BP, Gross_salary, Net_salary;
public Asst_Proffessor(int id, String name, String addr, String mail, String mob)
{
super(id, name, addr, mail, mob);
}
void computePay()
{
System.out.println("enter basic pay");
Scanner input = new Scanner(System.in);
BP = input.nextDouble();
Gross_salary = BP;
double DA, HRA, PF, Fund;
DA = (BP * 97 / 100);
HRA = (BP * 10 / 100);
PF = (BP * 12 / 100);
Fund = (BP * 0.1 / 100);
Net_salary = BP + DA + HRA - (PF + Fund);
System.out.println("Emp_Id:" + Emp_id);
System.out.println("Emp_name:" + Emp_name);
System.out.println("Address" + Address);
System.out.println("mail_Id:" + Mail_id);
System.out.println("Mobile_no:" + Mobile_no);
System.out.println("Grosspay:" + Gross_salary);
System.out.println("Netpay:" + Net_salary);
}
}
class Associate_Proffessor extends Employee
{
double BP, Gross_salary, Net_salary;
public Associate_Proffessor(int id, String name, String addr, String mail, String mob)
{ super(id, name, addr, mail, mob);
}
void computePay()
{
System.out.println("enter basic pay");
Scanner input = new Scanner(System.in);
```

```
BP = input.nextDouble();
Gross_salary = BP;
double DA, HRA, PF, Fund; DA = (BP * 97 / 100);
HRA = (BP * 10 / 100); PF = (BP * 12 / 100);
Fund = (BP * 0.1 / 100);
Net_salary = BP + DA + HRA - (PF + Fund);
System.out.println("Emp_Id:" + Emp_id);
System.out.println("Emp_name:" + Emp_name);
System.out.println("Address" + Address);
System.out.println("mail_Id:" + Mail_id);
System.out.println("Mobile_no:" + Mobile_no);
System.out.println("Grosspay:" + Gross_salary);
System.out.println("Netpay:" + Net_salary);
}
}
class Proffessor extends Employee
{
double BP, Gross_salary, Net_salary;
public Proffessor(int id, String name, String addr, String mail, String mob)
{
super(id, name, addr, mail, mob);
}
void computePay()
{
System.out.println("enter basic pay");
Scanner input = new Scanner(System.in); BP = input.nextDouble();
Gross_salary = BP;
double DA, HRA, PF, Fund; DA = (BP * 97 / 100);
HRA = (BP * 10 / 100);
PF = (BP * 12 / 100);
Fund = (BP * 0.1 / 100);
Net_salary = BP + DA + HRA - (PF + Fund);
System.out.println("Emp_Id:" + Emp_id);
System.out.println("Emp_name:" + Emp_name);
System.out.println("Address" + Address);
System.out.println("mail_Id:" + Mail_id);
System.out.println("Mobile_no:" + Mobile_no);
System.out.println("Grosspay:" + Gross_salary);
System.out.println("Netpay:" + Net_salary);
}
```

```
}
public class Main
{
public static void main(String[] args)
{
Programmer p = new Programmer(10, "AAA", "xxx", "aaa_xxx@gmail.com",
"1111111111");
System.out.println("....Programmer");
p.computePay();
Asst_Proffessor Ap = new Asst_Proffessor(20, "BBB", "yyy", "bbb_yyy@gmail.com",
"2222222222");
System.out.println("....Asst_Proffessor");
Ap.computePay();
Associate_Proffessor As = new Associate_Proffessor(10, "CCC", "zzz", "ccc_zzz@gmail.com",
"3333333333");
System.out.println("....Associate_Proffessor");
As.computePay();
Proffessor Pf = new Proffessor(40, "DDD", "www", "ddd_www@gmail.com", "4444444444");
System.out.println("....Proffessor");
Pf.computePay();
}
}
```

OUTPUT:

```
Programmer --------------------      ----------------------
EnterBasicPay:5000Emp_Id: 10Emp_Name: AAAAddress:xxx Mail_Id:
aaa_xxx@gmail.comMobileNumber:1111111111 GrossPay:10350.0
NetPay:9745.0

AssistantProfessor  ------------------     ----------------
EnterBasicPay:10000Emp_Id: 20Emp_Name: BBBAddress:yyy Mail_Id:
bbb_yyy@gmail.comMobileNumber:2222222222 GrossPay:10000.0
NetPay:19490.0

AssociateProfessor  ------------------      ----------------
EnterBasicPay:15000Emp_Id: 30Emp_Name: CCCAddress:zzz Mail_Id:
ccc_zzz@gmail.comMobileNumber:3333333333 GrossPay:15000.0
NetPay:29235.0
Professor  ------------------      ----------------
EnterBasicPay:20000Emp_Id: 40Emp_Name: DDDAddress:www Mail_Id:
ddd_www@gmail.comMobileNumber:4444444444 GrossPay:20000.0
NetPay:38980.0
```

Result:

Thus the java program to perform inheritance in java was executed and output is obtained.

Ex.No:04	**ABSTRACT CLASS**
Date:	

AIM:

To write and implement a java program for abstract class to find area of rectangle, triangle and circle.

ALGORITHM:

Step 1: Start the program.

Step 2: Declare the abstract class named “Shapes”.

Step 3: Write a class for rectangle and declare printArea() method to print the area.

Step 4: Write a class for tritangle and declare printArea() method to print the area.

Step 5: Write a class for circle and declare printArea() method to print the area.

Step 6: Create an object for the abstract class called “obj” and call the necessary methods to print all the area.

Step 7: Stop the program.

PROGRAM:

```
import java.util.*; abstract class Shapes
{
double a, b;
void printArea() {}
}
class Rectangle extends Shapes { void printArea()
{
System.out.println("\t\t Calculating Area of Rectangle");
Scanner input = new Scanner(System.in);
System.out.print("Enter length: ");
a = input.nextDouble();
System.out.print("Enter breadth: ");
b = input.nextDouble();
double area = a * b;
System.out.println("Area of Rectangle: " + area);
}
}
class Triangle extends Shapes
{
void printArea()
{
System.out.println("\t\tCalculating Area of Triangle");
Scanner input = new Scanner(System.in);
System.out.print("Enter height: "); a = input.nextDouble();
System.out.print("Enter breadth: "); b = input.nextDouble();
double area = 0.5 * a * b;
System.out.println("Area of Triangle: " + area);
}
}
class Circle extends Shapes
{
void printArea()
{

System.out.println("\t\t Calculating Area of Circle");
Scanner input = new Scanner(System.in);
System.out.print("Enter radius: ");
a = input.nextDouble(); double area = 3.14 * a * b;
System.out.println("Area of circle: " + area);
}
}
class abstractclassDemo
{
public static void main(String[] args)
{
Shapes obj;
obj = new Rectangle();
obj.printArea();
obj = new Triangle();
obj.printArea();
obj = new Circle();
obj.printArea();
}
}
```

OUTPUT:

```
D:\>javac AbstractClassDemo.javaD:\>javac AbstractClassDemo.java

Calculating Area of Rectangle Enter length:10
Enter breadth:20
Area of Rectangle:200.0 Calculating Area of Triangle Enter height:30
Enter breadth:25
Area of Rectangle: 375.0 Calculating Area of Circle Enter radius:10
Area of Circle: 314.0
```

Result:

Thus the java program to implement an abstract class to find area of rectangle, triangle and circle was executed and the output is obtained

Ex.No:05	INTERFACES IN JAVA
Date:	

AIM:

To write and implement a java program using interface to find area of rectangle, triangle and circle.

ALGORITHM:

Step1: Start the program.

Step2: Declare the interface named “Shapes”.

Step 3: Write a class for rectangle which implements shape and declare printArea() method to print the area.

Step 4: Write a class for triangle and declare printArea() method to print the area.

Step5: Write a class for circle and declare printArea() method to print the area.

Step6: Stop the program.

PROGRAM:

```
interface Shape
{
void input();
void area();
}
class Circle implements Shape
{
int r = 0;
double pi = 3.14, ar = 0;
@Override
public void input()
{
r = 5;
}
@Override public void area()
{
ar = pi * r * r;
System.out.println("Area of circle:"+ar);
}
}
class Rectangle extends Circle
{
int l = 0, b = 0; double ar;
public void input()
{
super.input();
l = 6;
b = 4;
}
public void area()
{
super.area();
ar = l * b;
System.out.println("Area of rectangle:"+ar);
}
}
public class Demo1
{
public static void main(String[] args)
{
Rectangle obj = new Rectangle();
obj.input();
obj.area();
}
}
```

OUTPUT:

```
Area of circle:78.5

Area of rectangle:24.0
```

Result:

Thus the java program to implement an interface to find area of rectangle, triangle and circle was executed and the output is obtained

Ex.No:06	**USER DEFINED EXCEPTION HANDLING**
Date:	

AIM:

To write and implement a java program for user defined exception handling.

ALGORITHM:

Step1: Start the program.

Step2: create a class MyException that extends Exception

Step3: Declare a class Clock should have hour and minute.

Step4: Write a class called exception that should handle any exception in the class clock.

Step5: Throws the exception by using MyException class

Step6: Stop the program.

PROGRAM:

```
import java.lang.Exception;
import java.io.*;
import java.util.*;
class MyException extends Exception
{
MyException(String msg)
{
super(msg);
}
}
class Clock
{
private int hour;
private int minute;
public void input() throws IOException
{
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
System.out.println("\nEnter the time in hh:mm format");
String str = br.readLine();
StringTokenizer tokn = new StringTokenizer(str, ":");
String h = tokn.nextToken();
String m = tokn.nextToken(); hour = Integer.parseInt(h);
minute = Integer.parseInt(m); try
{
System.out.println("Hour:" + hour);
if ((hour < 0) || (hour > 24)) throw new MyException("FatalError:InvalidHour");
}
catch (MyException e)
{
System.out.println(e.getMessage());
}
try
{
System.out.println("Minute:" + minute);
if ((minute < 0) || (minute > 59))
throw new MyException("Fatalerror: InvalidMinute");
}
catch (MyException e)
{
```

```
System.out.println(e.getMessage());
}
}
}
class ClockDemo
{
public static void main(String[] args) throws IOException
{
Clock c = new Clock(); c.input();
}
}
```

OUTPUT:

```
Enter the time in hh:mm format 25:80
Hour:25 FatalError: Invalid HourMinute:80
Fatalerror: Invalid Minute

Enter the time in hh:mm format10:70
Hour:10M inute:70
Fatalerror: Invalid Minute
```

Result:

Thus the java program to implement a user defined exception handling was executed and the output is obtained.

Ex.No:07	**MULTITHREADING IN JAVA**
Date:	

AIM:

To write a java Program that implements a multithreaded application that has three threads.

ALGORITHM:

Step1: Start the program.

Step2: Create a class called Number Generate.

Step3: Create a class called thread

Step4: Use thread methods to call, run the threads.

Step5: Write in class “TestNumber” to call the thread.

Step6: Stop the program.

PROGRAM:

```
import java.util.*; class NumberGenerate
{
private int value; private boolean flag;
public synchronized void put()
{
while (flag)
{
try
{
wait();
}
catch (InterruptedException e) {}
}
flag = true;
Random random = new Random(); this.value = random.nextInt(10);
System.out.println("The generated Numberis:" + value);
notifyAll();
}
public synchronized void get1()
{
while (!flag)
{
try
{
wait();
}
catch (InterruptedException e) {}
}
if (value % 2 == 0)
{
System.out.println("Second is executing now"); int ans = value * value;
System.out.println(value + "is evenNumber and its square is" + ans);
}
flag = false; notifyAll();
}
public synchronized void get2() { while (!flag)
{
try
```

```
{
wait();
}
catch (InterruptedException e) {}
}
if (value % 2 != 0)
{
System.out.println("Third thread is executing now...");
int ans = value * value * value;
System.out.println(value + "is OddNumber and the cube is:" + ans);
}
flag = false; notifyAll();
}
}
public class TestNumber
{
public static void main(String[] args)
{
final NumberGenerate obj = new NumberGenerate();
Thread ProducerThread = new Thread()
{
public void run()
{
for (int i = 1; i <= 6; i++)
{
System.out.println("Main thread Started...");
obj.put();
try
{
Thread.sleep(1000);
}
catch (InterruptedException e) {}
}
}
}
Thread consumerThread1 = new Thread();
public void run()
{
for (int i = 1; i <= 3; i++)
{
```

```
obj.get1();
try {
Thread.sleep(2000);
}
catch (InterruptedException e) {}
}
}
}
Thread consumerThread2 = new Thread() public void run()
{
for (int i = 1; i <= 3; i++)
{
obj.get2();
try {
Thread.sleep(3000);
}
catch (InterruptedException e) {}
}
}
}
ProducerThread.start(); consumerThread1.start(); consumerThread2.start();
}
}
```

OUTPUT:

```
Main thread Started...

The generated Numberis:4 Main thread Started...

The generated Numberis:9 Main thread Started...

The generated Numberis:9 Third thread is executing now...

9is OddNumber and the cube is:729 Main thread Started...

The generated Numberis:2 Second is executing now

2is evenNumber and its square is4 Main thread Started...

The generated Numberis:9 Main thread Started...

The generated Numberis:1 Third thread is executing now...

1is OddNumber and the cube is:1
```

RESULT:

Thus the java program for the implementation multithreading was executed and output was obtained.

Ex.No:08	**FILE OPERATION**
Date:	

AIM:

To write a java program that reads a filename from the user, displays information about whether the file exists, whether the file is readable or writable, the type of the file and the length of the file in bytes.

ALGORITHM:

Step 1: Start the program.

Step 2: Import the header file Scanner to read the files.

Step 3: Create a class called File Demo

Step 4: Get the input by using Scanner class

Step 5: Create an object for file class

Step 6: Declare the file objects to find the length and type of the file

Step 7: Stop the program.

PROGRAM:

```
import java.util.Scanner; import java.io.File;
class FileDemo
{
public static void main(String args[])
{
System.out.println("Enter the name of the file");
Scanner input = new Scanner(System.in);
Strings = input.nextLine();
Filef1 = new File(s);
System.out.println("                    ");
System.out.println("File name:" + f1.getName());
System.out.println("Path:" + f1.getPath());
System.out.println("Abs Path:" + f1.getAbsolutePath());
System.out.println("Thefileis:" + (f1.exists() ? "Exists" : "DoesnotExists"));
System.out.println("Isfile:" + f1.isFile());
System.out.println("Is Directory:" + f1.isDirectory());
System.out.println("Is Readable:" + f1.canRead());
System.out.println("Is Writable:" + f1.canWrite());
System.out.println("Is Absolute:" + f1.isAbsolute());
System.out.println("File Size:" + f1.length() + "bytes");
System.out.println("IsHidden:" + f1.isHidden());
}
}
```

OUTPUT:

```
Filename:FileDemo.javaP ath:FileDemo.java
Abs Path:D:\vimala\Materials\Java Pgms\FileDemo.javaThefile is:Exists Isfile:true
IsDirectory:falseI  s Readable:true IsWritable:true IsAbsolute:false FileSize:883bytesI
sHidden:false
```

RESULT:

Thus the java program for implementing file operations was executed successfully and the output was obtained.

Ex.No:09	**GENERIC METHOD IMPLEMENTATION**
Date:	

AIM:

To write a java program for implementing concept of generic method.

ALGORITHM :

Step 1: Create a generic method T max to find the maximum of given values.

Step 2: Use any pre-defined List of integers and characters as input.

Step 3: Assign the first element from the MyList to element.

Step 4: Check for next_element.

Step 5: If found Compare it with element. If greater assign it to element.

Step 6: Else retain element.

Step 7: Repeat step4 until no next_element available.

Step 8: Print the final maximum value element.

Step 9: Stop the program

PROGRAM:

```
import java.util.*;
public class Test
{
public static < T extends Object & Comparable < ? super T >> T max(Collection < ? extends T >
coll)
{
Iterator < ? extends T > MyList = coll.iterator();
T element = MyList.next(); while (MyList.hasNext())
{
T next_element = MyList.next();
if (next_element.compareTo(element) > 0) element = next_element;
}
return element;
}
public static void main(String args[])
{
List < Integer > ints = new ArrayList < Integer > (Arrays.asList(1, 55, 4, 3, 23, 12, 25, 9));
int max = Collections.max(ints);
System.out.println(ints);
System.out.println("MaximumValueis:" + max);
List < Character > chars = new ArrayList < Character > (Arrays.asList('a', 'e', 'i', 'o', 'u'));
char maxc = Collections.max(chars);
System.out.println(chars);
System.out.println("MaximumValueis:" + maxc);
}
}
```

OUTPUT:

```
D:\APA\Lab_Programs>javacTest.java D:\APA\La_Programs>javaTest

[1, 55, 4, 3, 23, 12, 25, 9]
MaximumValueis:55 [a, e, i, o, u] MaximumValueis:u
```

RESULT:

Thus the JAVA program for the implementation of generic method was executed and output was obtained.

Ex.No:10	**Develop applications using JavaFX controls, layouts and menus**
Date:	

AIM:

To write a java program for creating an applications using JavaFX controls, layouts and menus

ALGORITHM:

Step 1: Start the program.

S**tep 2**: Extend javaFX application and override start () method.

Step 3: Create a button and event for the button.

Step 4: Create a layouts, controls and menus add the button to it.

Step 5: Create the scene and prepare the stage.

Step 6: Create the main method.

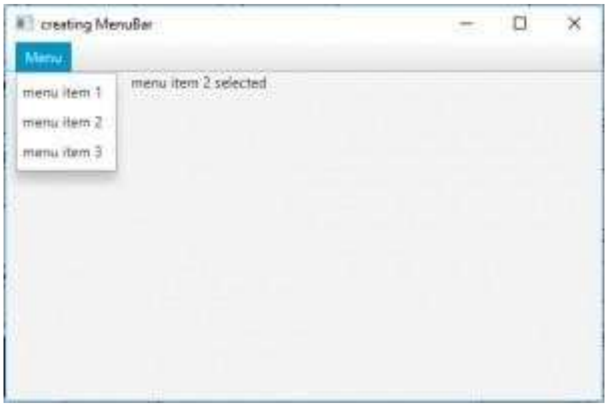
Step 7: Run the application to see the output.

PROGRAM:

```
import javafx.application.Application;
import  javafx.scene.Scene;
import  javafx.scene.control.Button;
import javafx.scene.layout.*;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.control.*;
import  javafx.stage.Stage;
import javafx.scene.control.Alert.AlertType;
import  java.time.LocalDate;
public class MenuBar_2 extends Application
{
public  void  start(Stage  s)
{
s.setTitle("creating MenuBar");
Menu m = new Menu("Menu");
MenuItem m1 = new MenuItem("menu item 1");
MenuItem m2 = new MenuItem("menu item 2");
MenuItem  m3 = new MenuItem("menu item3");
m.getItems().add(m1);
m.getItems().add(m2);
m.getItems().add(m3);
Label l = new Label("\t\t\t\t" + "no menu item selected");
EventHandler<ActionEvent>  event  = new EventHandler<ActionEvent>()
{
public void handle(ActionEvent e)
{
l.setText("\t\t\t\t" + ((MenuItem)e.getSource()).getText()  + " selected");
}
}
m1.setOnAction(event); m2.setOnAction(event);
m3.setOnAction(event);
MenuBar mb = new MenuBar();
mb.getMenus().add(m);
VBox vb = new VBox(mb,  l);
Scene sc = new Scene(vb,  500, 300);
s.setScene(sc); s.show();
}
public static void main(String args[])
{
```

```
launch(args);
}
}
```

OUTPUT:

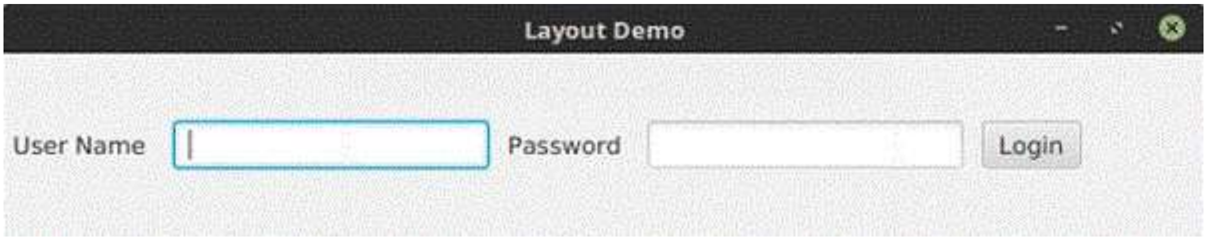


menu item 2 selected

PROGRAM :

```
package org.mano.example;
import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.PasswordField;
import javafx.scene.control.TextField;
import javafx.scene.layout.HBox;
import javafx.stage.Stage;
public class LayoutDemo extends Application
{
public static void main(String[] args)
{
Application.launch(args);
}
@Override
public void start(Stage stage) throws Exception
{
Scene scene = new Scene(createHBoxLayout(), 650, 100);
stage.setTitle("Layout Demo");
stage.setScene(scene);stage.show();
}
public HBox createHBoxLayout()
{
HBox hbox = new HBox(); hbox.setSpacing(10);
hbox.setPadding(new Insets(5));
hbox.setAlignment(Pos.CENTER_LEFT);
Label userLabel=new Label("User Name ");
Label passLabel=new Label("Password ");
TextField userTextField=new TextField();
PasswordField passwordField=new PasswordField();
Button loginButton=new Button("Login");
hbox.getChildren().addAll(userLabel,userTextField, passLabel,passwordField,loginButton);
return  hbox;
}
}
```

OUTPUT:



RESULT:

Thus the JAVA program for the implementation of generic method was executed and output was obtained.

Ex.No:11	**MINI PROJECT**
Date:	

AIM:

To write and implement a java program to develop a Calculator Application

ALGORITHM:

Step 1: Start the program

Step 2: Create the class structures.

Step 3: Declare the inputs required.

Step 4: Build the user input method.

Step 5: Declare the output value.

Step 6: Build the switch statement.

Step 7: Display the output using print function.

Step 8: Stop the program.

PROGRAM:

```
import java.awt.*;

import java.awt.event.*;

class MyCalc extends WindowsAdapter implements ActionListener
{
Frame f; Label l1;

Button  b1, b2, b3, b4, b5, b6, b7, b8, b9, b0;
Button badd, bsub, bmult, bdiv, bmod, bcalc, bcls, bpts, bneg, bback  ; double xd;

double num1,num2,check;

MyCalc()
{
f = new Frame(“MY CALCULATOR”);
l1= new Label(); l1.setBackground(Color.LIGHT_GRAY); l1.setBounds(50,50,260,60);

b1= new Button(“1”); b1.setBounds(50,340,50,50); b2= new Button(“2”);

b2.setBounds(120,340,50,50); b3= new Button(“3”);

b3.setBounds(190,340,50,50); b4= new Button(“4”); b4.setBounds(50,270,50,50); b5= new

Button(“5”); b5.setBounds(120,270,50,50); b6= new Button(“6”); b6.setBounds(190,270,50,50);

b7= new Button(“7”); b7.setBounds(50,200,50,50); b8= new Button(“8”);

b8.setBounds(120,200,50,50); b9= new Button(“9”); b9.setBounds(190,200,50,50); b0= new

Button(“0”); b0.setBounds(120,410,50,50); bneg= new Button(“+/-“);

bneg.setBounds(50,410,50,50); bpts= new Button(“.“); bpts.setBounds(190,410,50,50); bback=

new Button(“back“);

bback.setBounds(120,130,50,50); badd= new Button(“+“); badd.setBounds(260,340,50,50); bsub=

new Button(“-“); bsub.setBounds(260,270,50,50); bmult= new Button(“*“);

bmult.setBounds(260,200,50,50); bdiv= new Button(“/“); bdiv.setBounds(260,130,50,50); bmod=

new Button(“%“); bmod.setBounds(190,130,50,50); bcalc= new Button(“=“);

bcalc.setBounds(245,410,50,50); bclr= new Button(“CE“); bclr.setBounds(50,130,50,50);
b1.addActionListener(this); b2.addActionListener(this); b3.addActionListener(this);
b4.addActionListener(this); b5.addActionListener(this); b6.addActionListener(this);
b7.addActionListener(this); b8.addActionListener(this); b9.addActionListener(this);
b0.addActionListener(this); bneg.addActionListener(this); bpts.addActionListener(this);
```

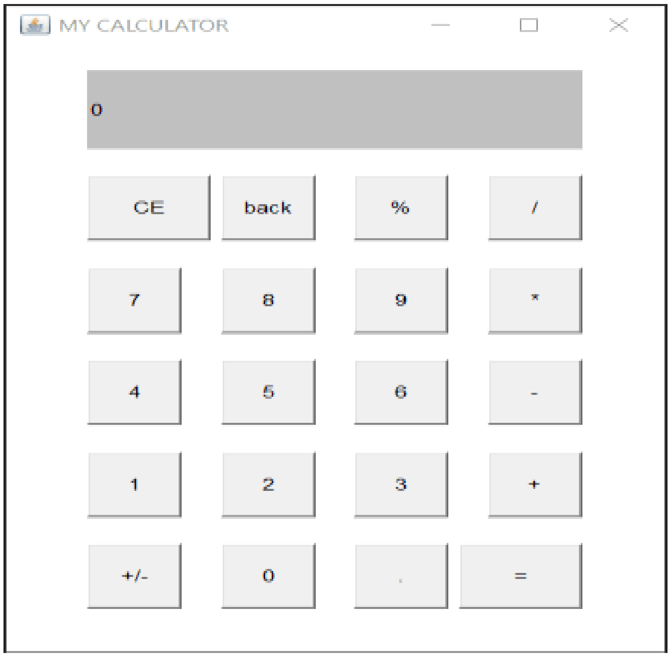
```
bback.addActionListener(this); badd.addActionListener(this); bsub.addActionListener(this);
bmult.addActionListener(this); bdiv.addActionListener(this); bmod.addActionListener(this);
bcalc.addActionListener(this); bclr.addActionListener(this); f.addWindowListener(this); f.add(l1);
f.add(b1);
f.add(b2);
f.add(b3);
f.add(b4);
f.add(b5);
f.add(b6);
f.add(b7);
f.add(b8);
f.add(b9);
f.add(b0);
f.add(badd);
f.add(bsub);
f.add(bmult);
f.add(bdiv);
f.add(bmod);
f.add(bcalc);
f.add(bclr);
f.add(bpts);
f.add(bneg);
f.add(bback);
f.setSize(360,500); f.setLayout(null); f.setVisible(true);
}
public void windowClosing(WindowEvent e)
{
f.dispose();
}
public void actionPerformed(ActionEvent e)
{
String z,zt; if(e.getSource()==b1)

{
zt=l1.getText(); z=zt+"1"; l1.setText(z);
}
if(e.getSource()==b2)
{
zt=l1.getText(); z=zt+"2"; l1.setText(z);

}
if(e.getSource()==b3)
{
zt=l1.getText(); z=zt+"3"; l1.setText(z);

}
if(e.getSource()==b4)
{
zt=l1.getText(); z=zt+"4"; l1.setText(z);

}
if(e.getSource()==b5)
{
zt=l1.getText(); z=zt+"5"; l1.setText(z);

}
if(e.getSource()==b6)
{
zt=l1.getText(); z=zt+"6"; l1.setText(z);

}
if(e.getSource()==b7)
{
zt=l1.getText(); z=zt+"7"; l1.setText(z);

}
if(e.getSource()==b8)
{
zt=l1.getText(); z=zt+"8"; l1.setText(z);

}
if(e.getSource()==b9)
{
zt=l1.getText(); z=zt+"9"; l1.setText(z);

}
if(e.getSource()==b0)
{
zt=l1.getText(); z=zt+"0"; l1.setText(z);

}
if(e.getSource()==bpts)
{
zt=l1.getText(); z=zt+"."; l1.setText(z);

}
```

```
if(e.getSource()==bneg)
{
zt=l1.getText(); z=zt+”-”; l1.setText(z);

}
if(e.getSource()==bback)
{
zt=l1.getText(); try

{
z=zt.subString(0,zt.length()-1);
}
catch(StringIndexOutOfBoundsException  f)
{
return ;
}
l1.setText(z);
}
if(e.getSource()==badd)
{
try
{
num1=Double.parseDouble(l1.getText());
}
catch(NumberFormatException f)
{
l1.setText(“Invalid Format”); return ;
}
z=” “;
l1.setText(z); check=1;
}
if(e.getSource()==bsub)
{
try
{
num1=Double.parseDouble(l1.getText());
}
catch(NumberFormatException f)
{
l1.setText(“Invalid Format”);
return ;
}
z=” “;
l1.setText(z);
check=2;
}
if(e.getSource()==bmult)
{
```

```
try
{
num1=Double.parseDouble(l1.getText());
}
catch(NumberFormatException f)
{
l1.setText(“Invalid Format”);
return ;
}
z=” “; l1.setText(“Invalid Format”);
 return ;
l1.setText(z); check=3;
}
if(e.getSource()==bdiv)
{
try
{
num1=Double.parseDouble(l1.getText());
}
catch(NumberFormatException f)
{
l1.setText(“Invalid Format”);
return ;
}
z=” “;
l1.setText(z); check=4;
}
if(e.getSource()==bmod)
{
try
{
num1=Double.parseDouble(l1.getText());
}
catch(NumberFormatException f)
{
l1.setText(“Invalid Format”);
return ;
}
z=” ”;
l1.setText(z);
check=5;
}
}
if(e.getSource()==bcalc)
{
try
{
num2=Double.parseDouble(l1.getText());
```

```
}
catch(Exception f)
{
l1.setText(“Enter Number First”); return;
}
if(check==1)
xd=num1+num2;
if(check==2)
xd=num1-num2;
if(check==3)
xd=num1*num2; if(check==4)
xd=num1/num2; if(check==5)
xd=num1%num2; l1.setText(String.valueOf(xd);
}
if(e.getSource()==bclr)
{
num1=0; num2=0; check=0; xd=0; z=” ”;
l1.setText(z);
}}
public static void main(String[] args)
{
new MyCalc();
}
}
```

OUTPUT:



RESULT:

Thus the java program for implementing file operations was executed successfully and the output was obtained.

VIVAQUESTIONS:

1. What is the use of looping statement.
2. What is the difference between dowhile and while
3. Give the ternary operators.
4. What are difference between break and continue?
5. Can a for statement loop indefintely?
6. Define Stack and Queue.
7. Explain about stack operations
8. What are the advantages of stack?
9. What are the disadvantages of stack?
10. Differentiate stack with queue.
11. What is the use of inheritance?
12. What is interface?
13. List the types of inheritance
14. Give the different types of visibility controls
15. What are the keywords used to inherit a class from base class?
16. What is abstract class?
17. Can abstract class have constructors in Java?
18. Can abstract class be final in Java?
19. Is it necessary for abstract class to have abstract method?
20. Difference between abstract class and interface in Java?
21. What methods would a class that implements the java.lang.Char Sequence interface have to implement?
22. What is an exception?
23. How the exceptions are handled in java? OR Explain exception handling mechanism in java?
24. What is the difference between error and exception in java?
25. What is unreachable catch block error?
26. Explain the hierarchy of exceptions in java?
27. What are runtime exceptions in java. Give example?
28. What is Thread in java?
29. What is difference between Process and Thread in java?
30. How to implement Threads in java?

31. What is difference between starting thread with run() and start()method?

32. How threads communicate between each other?

33. What is dead lock in multithreading?

34. What is the difference between System.out, System. errandSystem.in?

35. Which is the abstract parent class of FileWriter?

36. What is PrintStream and PrintWriter?

37. What is File class?

38. What is Random Access File?