CMR TECHNICAL CAMPUS

Kandlakoya(V),Medchal (M),Medchal District (T.S)

An UGC Autonomous Institute

Accredited by NBA and NAAC with A Grade

Approved by AICTE, New Delhi and Affiliated to JNTU, Hyderabad

JAVA PROGRAMMING

Laboratory Manual

(R20)

Department Of

Computer Science and Engineering (Data Science)

Year: II Year IIISem

S.No	Experiment Name
1	Use Ecilipse or Netbean platform and acquaint with the various menus. Create a test project, add a test class and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.
2	Write a Java Program to create an abstract class named shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the Class Shape. Each one of the classes contains only the method printArea() that prints the area of the given Shape.
3	Write a Java Program that implements Bubble Sort algorithm for sorting in descending order and also show the number of interchanges occurred for the given set of integers.
4	Write a Java Program that implements Quick Sort algorithm for sorting a list of names in ascending order.
5	Write a Java Program for the following: Create a doubly linked list of elements. Delete a given element from the above list Display the contents of the list after deletion.
6	Write a Java Program to list all the files in a directory including files present in all its subdirectories.
7	Write a Program that creates User Interface to perform Integer Divisons. The user enters two numbers in text fields, Num1 and Num2.The division of Num1 and Num2 is displayed in the result field when the divide button clicked. If Num1 or Num2 were not integer, the program would throw a Number Format Exception, If Num2 is Zero, and the program would throw an Arthematic Exception. Display the Exception in message dialog box.
8	Write Java Program that implements a multithread application that has three threads. First thread generates random integer for every second and if the value is even, second thread computes the square of number and prints. If the value is odd, the third thread will print the value of cube of number.
9	Write a Java Program on Producer consumer problem by using the concept of interthreadcommunication.
10	a)Develop an Applet in java that displays a Simple Message b) Develop an Applet in java that receives an integer in one TextField, and computes its Factorial value and returns it in another textfield, when button named “Compute” is

	clicked.
11	Write a Java Program that works as simple calculator .Use grid layout to arrange buttons for the digits and for the +,-,*,% operations .Add text field to display the results, Handle any possible exceptions like divide by zero. Suppose that a table named Table.txt is stored in atext file. The first line in the file is the header, and remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using Labels in GridLayout.
12	Write a Java Program that simulates a Traffic Light. The program lets the user select one of three lights::red, yellow or Green with radio buttons. On selecting radio button, an appropriate message with “stop” or “Ready” or “GO” should appear above the buttons in selected color. Intially ,there is no message shown.
13	Write a Java Program that handles all mouse events and shows the event name at the center of the window when the mouse event is fired.(Use Adapter Classes)
14	Write a java program that loads names and phone numbers from the text file where data is organized as one line per record and each field in record are separated by a tab(\t).It takes a name or phone number as input and prints corresponding other value from hash table(hint: use Hash Table)
15	Suppose that a table named Table.txt is stored in a text file. The First line in the file is the header, and the remaining lines correspond to rows in table. The elements are separated by commas. Write java program to display the table using Labels in Grid Layout.

Installation of Java software:

Steps:

1. Download JDK7.0 from www.java.sun.com/downloads
2. Install JDK by double clicking on setup file
3. Follow the installation wizard.
4. Upon successful installation the default location where java is installed is **C:\Program Files\Java directory.**
5. To compile a java class we have to use javac (java compiler)
6. To execute java program we have to use jvm (java virtual machine).
7. To use javac and jvm we have to configure environment variables.

Environment Variables

PATH:

- This environment variable is to locate the compiler.
- Right click MYCOMPUTER and find a properties tab, Click on Properties Tab and find Advanced Tab, Click on it to see environment variables.
- In this create a new environment variable called **PATH** and copy the location of java compiler as value to it.

Default location:

$PATH= C:\Program Files\Java\Jdk1.6.0\bin;

CLASSPATH:

This environment variable will help the jvm to find the runtime environment of java i.e. all predefined classes and interfaces can be located using this variable. The location for runtime environment is JRE folder.

$CLASSPATH= C:\Program Files\Java\Jre\bin;

WEEK-1

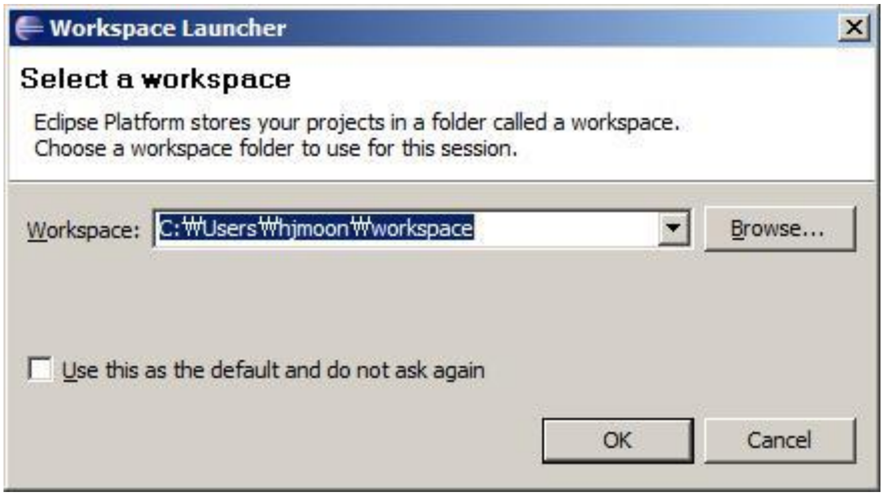
1.Use Eclipse or Netbeans platform and acquaint with the various menus. Create a test project, and a test class and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.

AIM:

To create and test java projects in Eclipse and Netbean platform

Initializing Eclipse

When you develop Java applications in Eclipse, it stores all the created files in a directory called "workspace". When Eclipse is run for the first time, it will ask you where you want the workspace to be placed:



Workspace Launcher

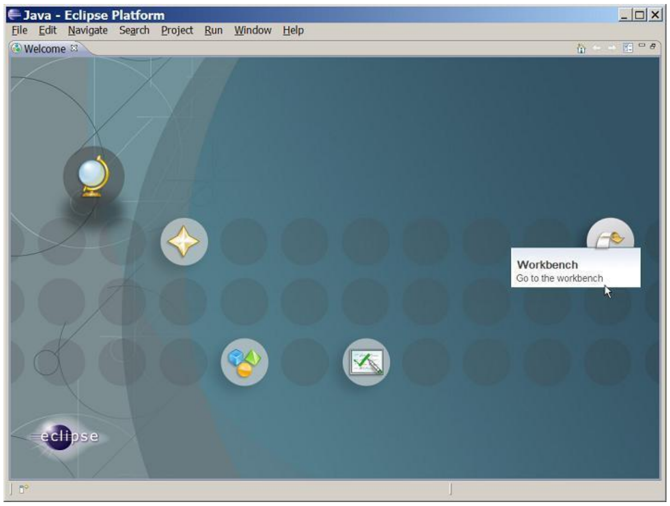
Select a workspace

Eclipse Platform stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

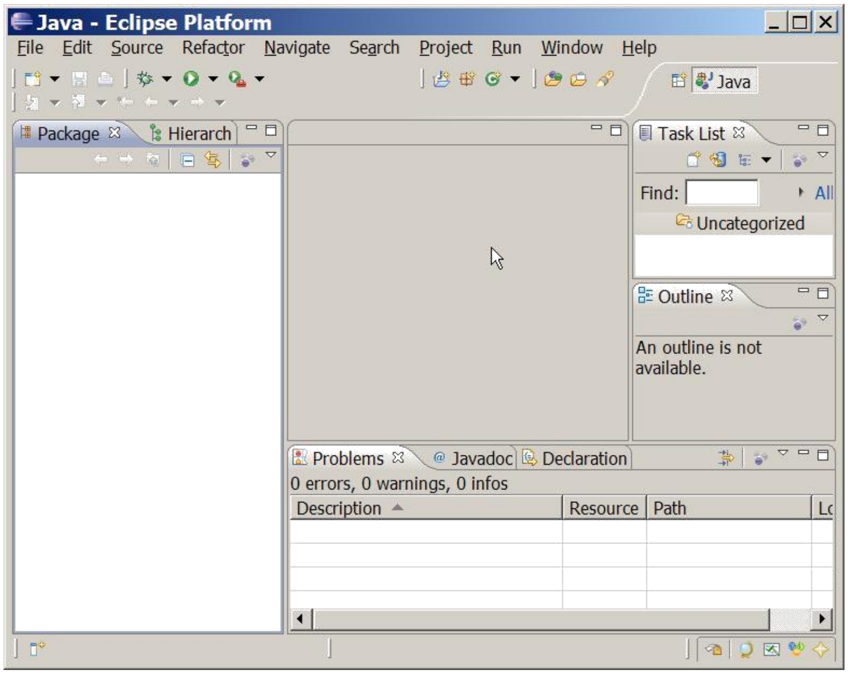
Workspace: C:₩Users₩hjmoon₩workspace

□ Use this as the default and do not ask again

You can just use the default location or specify your preferred location. To avoid getting asked this question every time you start Eclipse, check "Use this as the default and do not ask again" option and press "OK" button. Once Eclipse finishes its startup process, youwillseethefollowingwelcome window:



Click the "Workbench" icon on the right, which will lead you to the main Eclipse window:

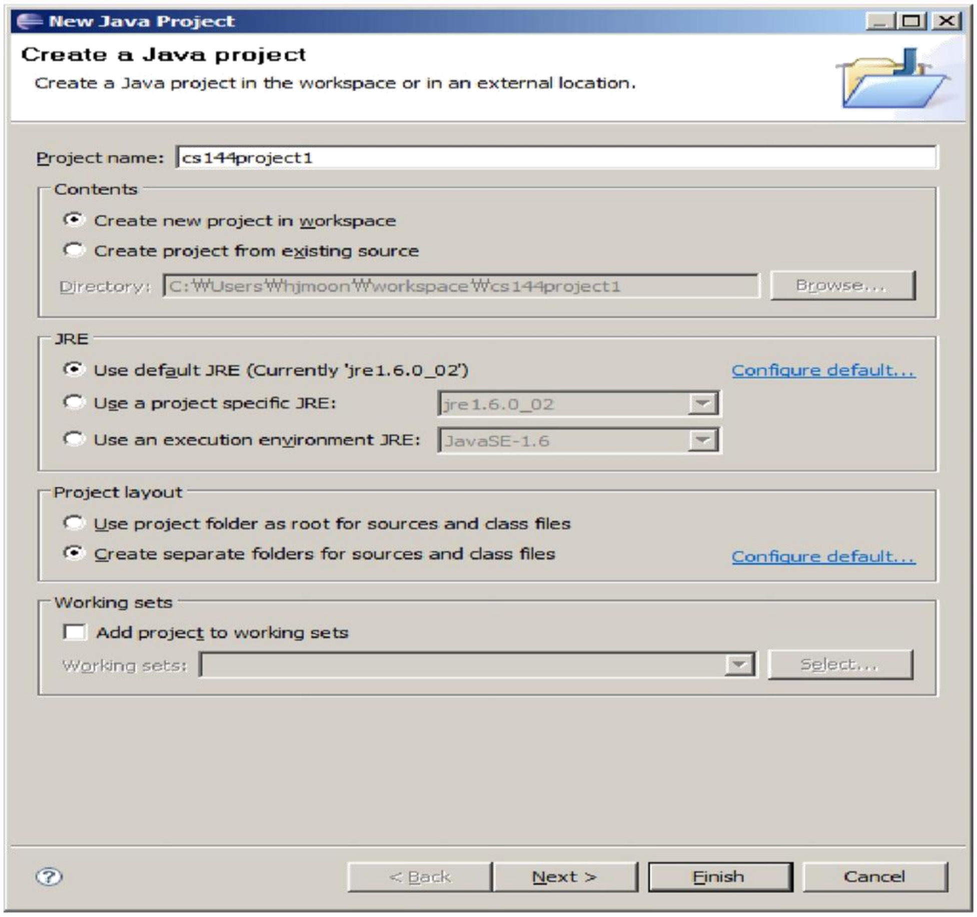


An outline is not available.

0 errors, 0 warnings, 0 infos

Creating a Project

Now that you've got Eclipse up and running, it's time to create your first Java project. To do this, you'll want to go File -> New -> Java Project. After doing so, you'll see a window like the following:



Create a Java project

Create a Java project in the workspace or in an external location.

Project name: cs144project1

Contents

● Create new project in workspace

○ Create project from existing source

Directory: C:₩Users₩hjmoon₩workspace₩cs144project1

JRE

● Use default JRE (Currently 'jre1.6.0_02')

○ Use a project specific JRE: jre1.6.0_02

○ Use an execution environment JRE: JavaSE-1.6

Project layout

○ Use project folder as root for sources and class files

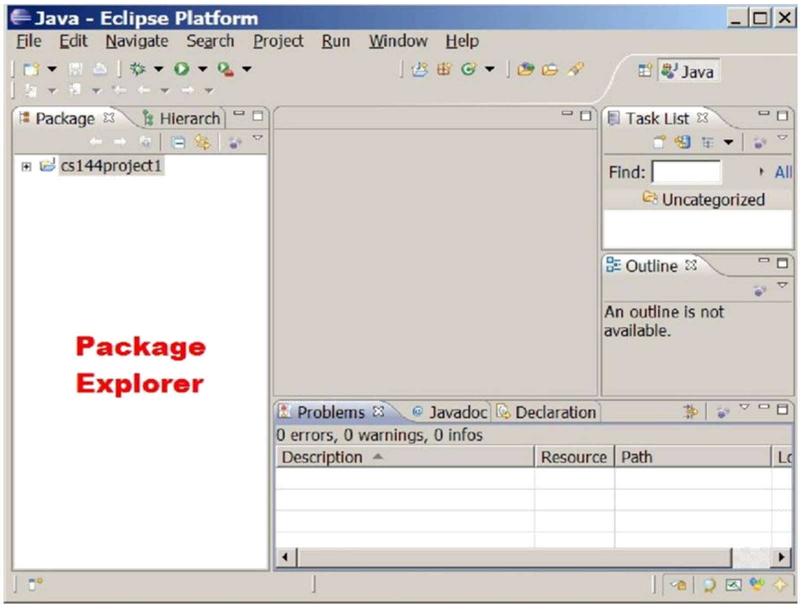
● Create separate folders for sources and class files

Working sets

□ Add project to working sets

Working sets:

Type your project name (say, cs144project1) in the "Project name" field and click Finish. Then the name of your newly created project will appear on the left side of the Eclipse window (this part of the window is called "Package explorer pane"



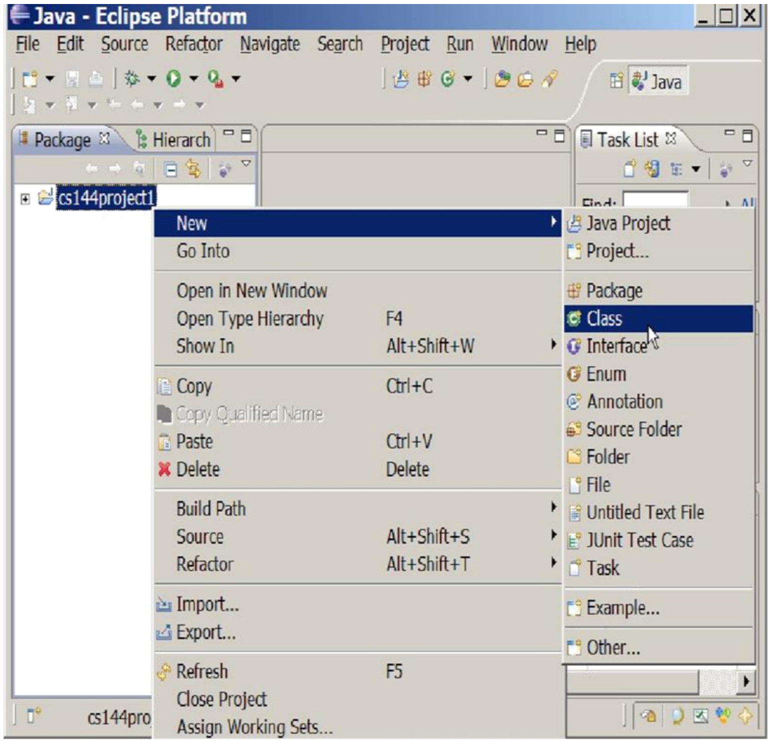
An outline is not available.

0 errors, 0 warnings, 0 infos

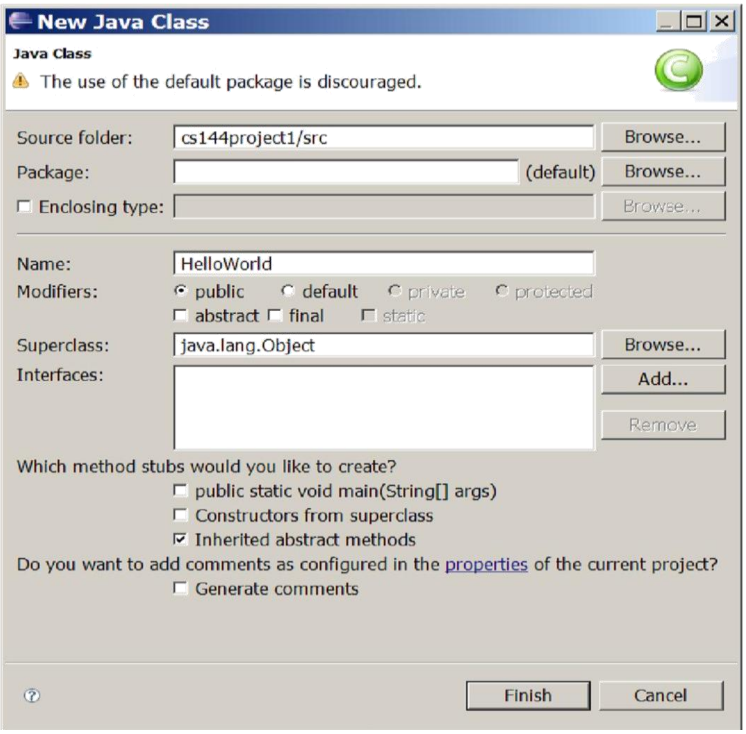
As you create more projects in Eclipse, other project names will appear in the Package explorer pane and you will be able to switch between your projects by clicking the name of a project.

Adding a New File to a Project

Now that you've created your first project, you now want to create a new Java file (with.java extension) and add it into your project. To create a new Java file, right click on the name of yourproject (cs144project1) in the Package explorer pane and select New -> Class as follows:



This command will show you a window that looks like the following:



New Java Class

Java Class

⚠ The use of the default package is discouraged.

Source folder: cs144project1/src

Package: (default)

□ Enclosing type:

Name: HelloWorld

Modifiers: ● public ○ default ○ private ○ protected

□ abstract □ final □ static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

□ public static void main(String[] args)

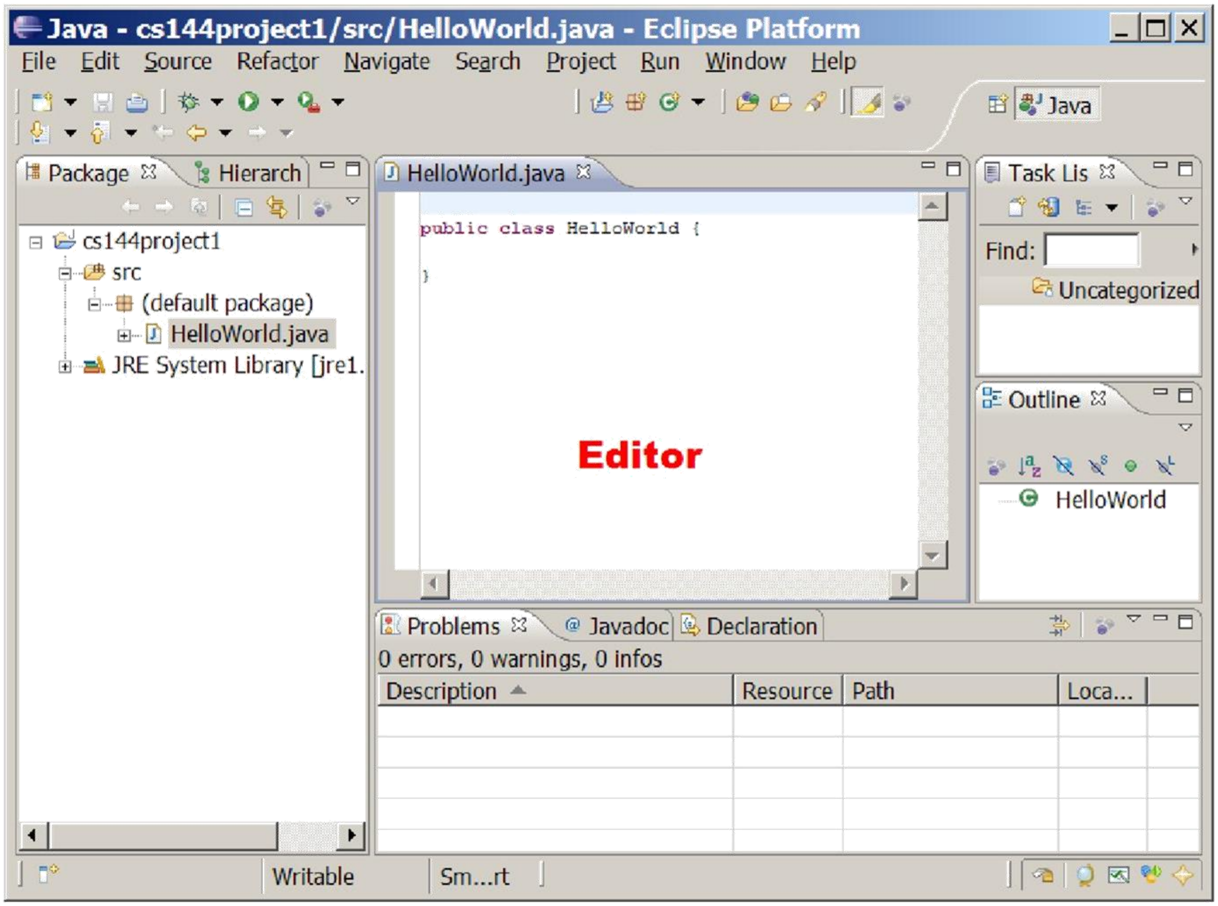
□ Constructors from superclass

☑ Inherited abstract methods

Do you want to add comments as configured in the properties of the current project?

□ Generate comments

In the "Name: " section provide the name of the file (or the class) you want to create, HelloWorld, and click "Finish" button



Java - cs144project1/src/HelloWorld.java - Eclipse Platform

- cs144project1
  - src
    - (default package)
      - HelloWorld.java
  - JRE System Library [jre1.

```
public class HelloWorld {

}
```

Find:

Uncategorized

HelloWorld

0 errors, 0 warnings, 0 infos

Description	Resource	Path	Loca...

Writable

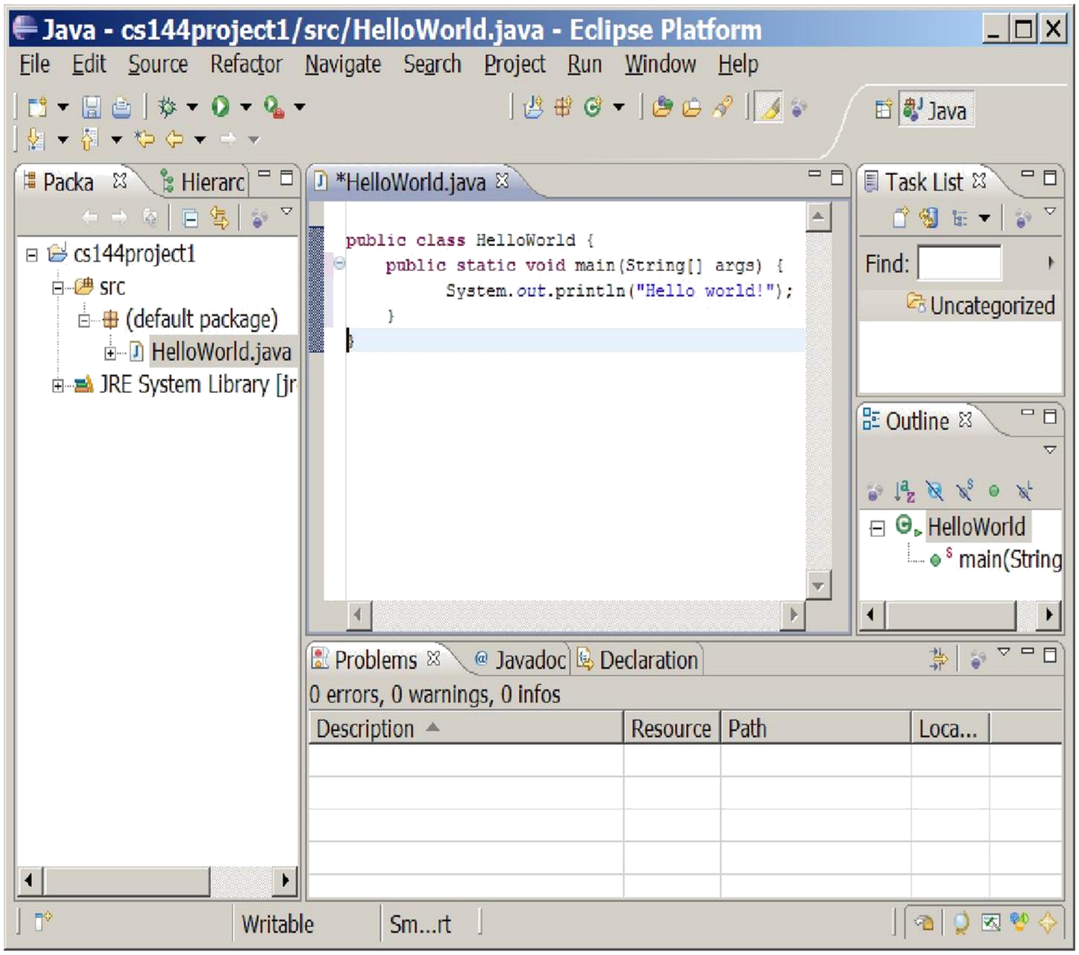
Sm...rt

Congratulations! Now you have created your first Java code in eclipse. As you can see from the Package explore pane, your project now includes HelloWorld.java file. The "Editor pane" to the right of the Package explorer pane shows the actual content of the HelloWorld.java file, which simply declares HelloWorld as a public class. You can edit the content of the Java code inside the Editor pane.

Saving, compiling, and running Java code

Now let us learn how to code, compile and run a Java program in Eclipse. First copy and paste the following method into the HelloWorld class definition:

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

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



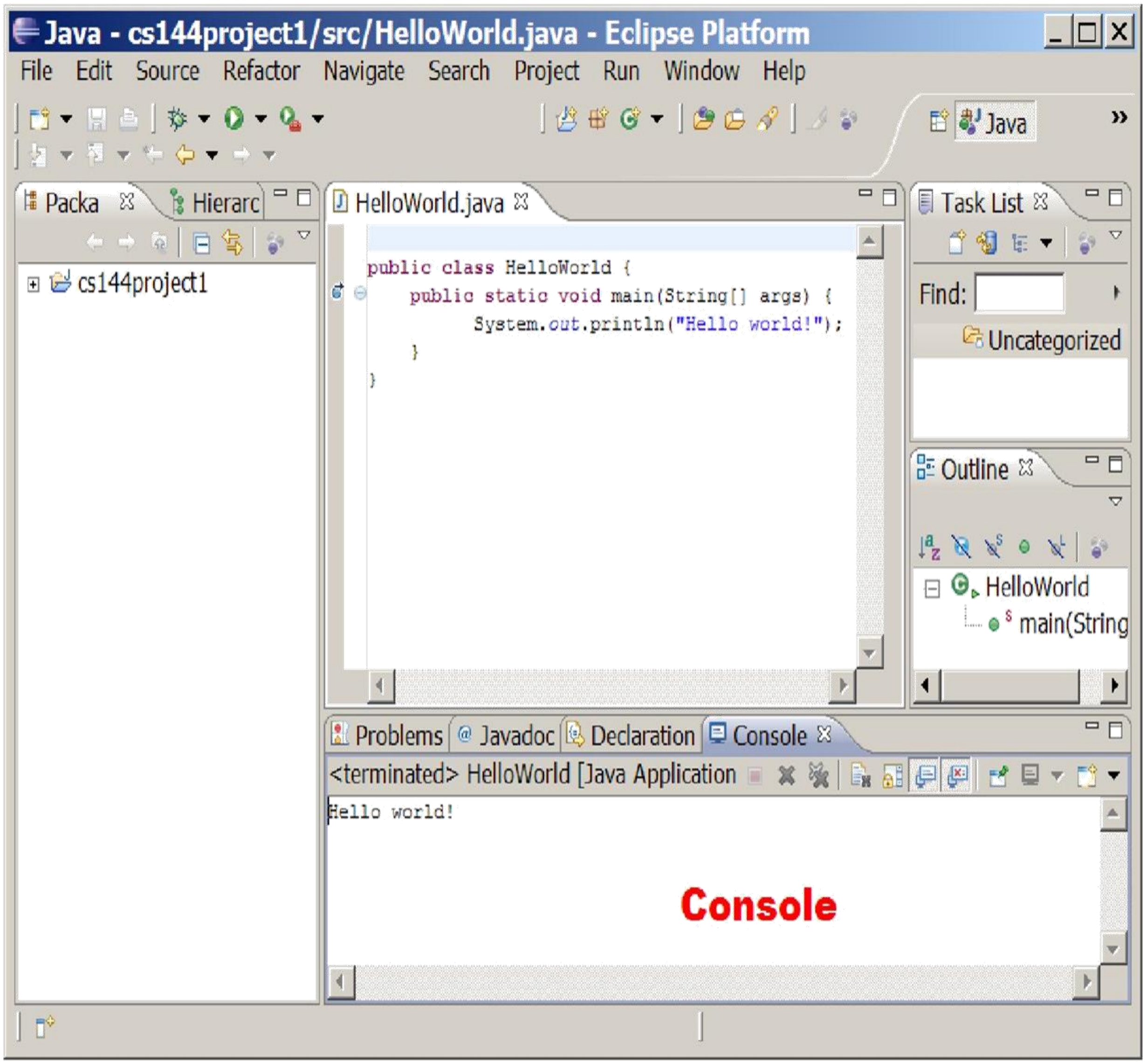
```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello world!");
    }
}
```

0 errors, 0 warnings, 0 infos

Now save the file by selecting File->Save, or pressing Ctrl-S (Option-S on Mac). When you save a Java file, Eclipse will automatically *compile* the file also, so that you don't need to compile it later when you want to run it.

Now that your code has been saved and compiled, you can run your program by selecting Run -> Run, or by pressing Ctrl-F11 (Option-F11 on Mac) or by clicking on the "Run"

button near the top of the window. Once your program finishes running, you will be able to see the output of your program by selecting the "Console tab" at the bottom of the window.



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

<terminated> HelloWorld [Java Application

```
Hello world!
```

Quitting Eclipse

You can exit eclipse by using any of the following alternatives:

- Hit the X in the upper right corner
- Select File -> Exit

Now you have learned the very basic functionality of Eclipse. Eclipse supports many more functionalities than what you just learned, including integrated debugging and automatic code completion and method lookup, etc. Please read online Eclipse manual to learn more about Eclipse.

Netbeans

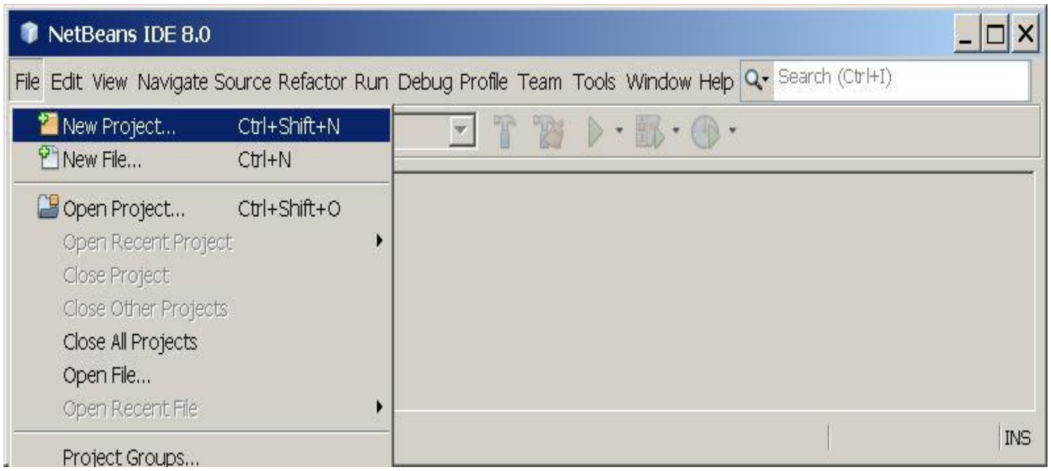
Create an IDE Project

To create an IDE project:

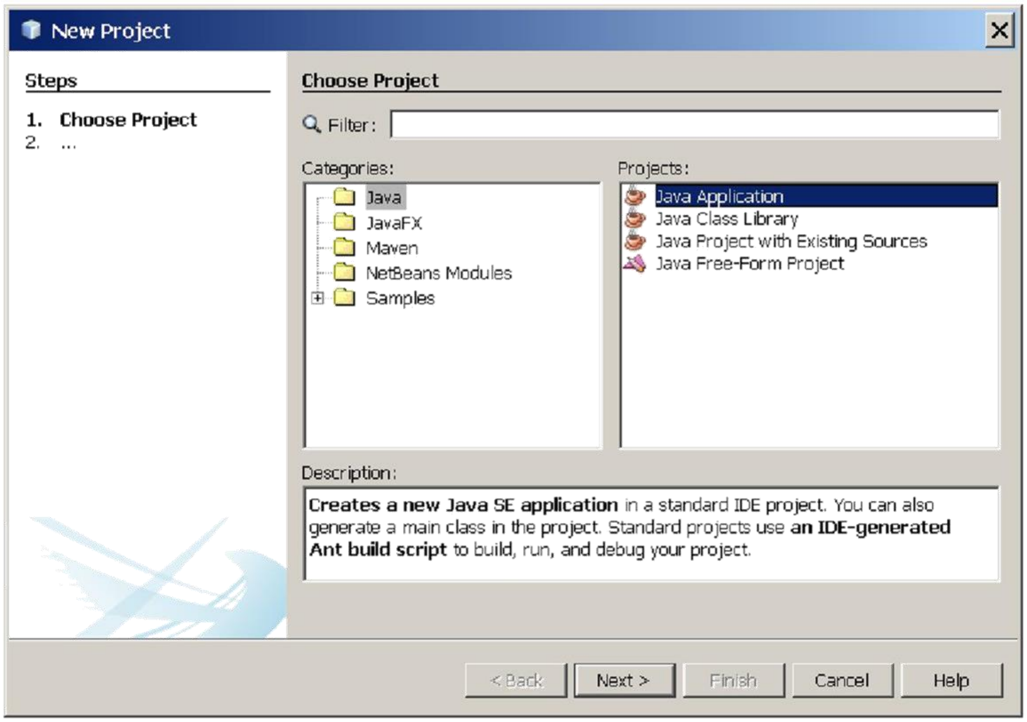
1. Launch the NetBeans IDE.

   - On Microsoft Windows systems, you can use the NetBeans IDE item in the Start menu.
   - On Solaris OS and Linux systems, you execute the IDE launcher script by navigating to the IDE's bin directory and typing ./netbeans.
   - On Mac OS X systems, click the NetBeans IDE application icon.

2. In the NetBeans IDE, choose **File | New Project...**.



1. NetBeans IDE with the File | New Project menu item selected.
2. In the **New Project** wizard, expand the **Java** category and select **JavaApplication** as shown in the following figure:

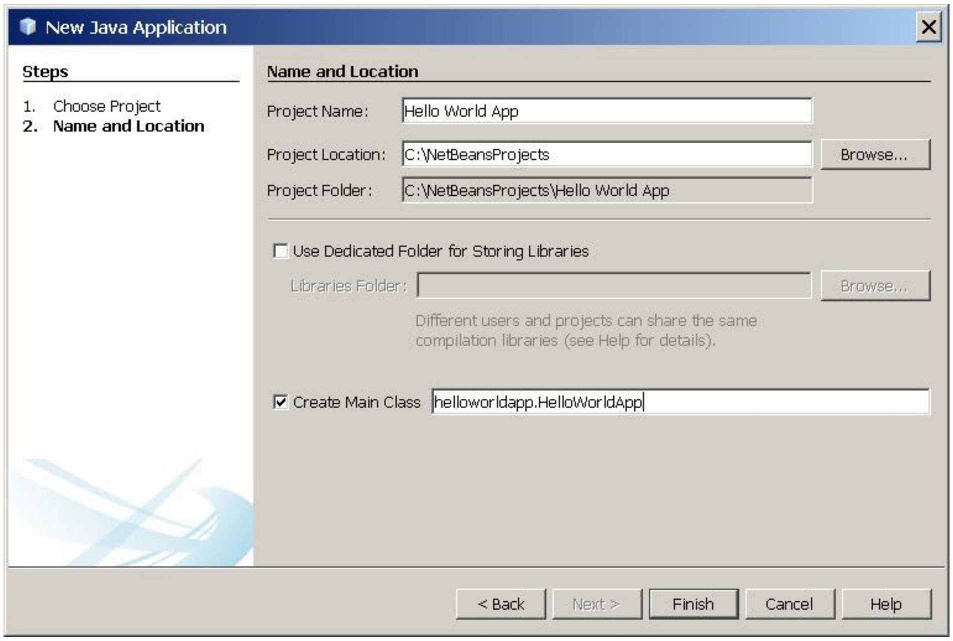


Creates a new Java SE application in a standard IDE project. You can also generate a main class in the project. Standard projects use **an IDE-generated Ant build script** to build, run, and debug your project.

1. NetBeans IDE, New Project wizard, Choose Project page.

2. In the **Name and Location** page of the wizard, do the following (as shown in the figure below):
   - In the**Project Name**field, type Hello World App.

   - In the**Create Main Class**field, type helloworldapp.HelloWorldApp.



Project Name: Hello World App

Project Location: C:\NetBeansProjects

Project Folder: C:\NetBeansProjects\Hello World App

☐ Use Dedicated Folder for Storing Libraries

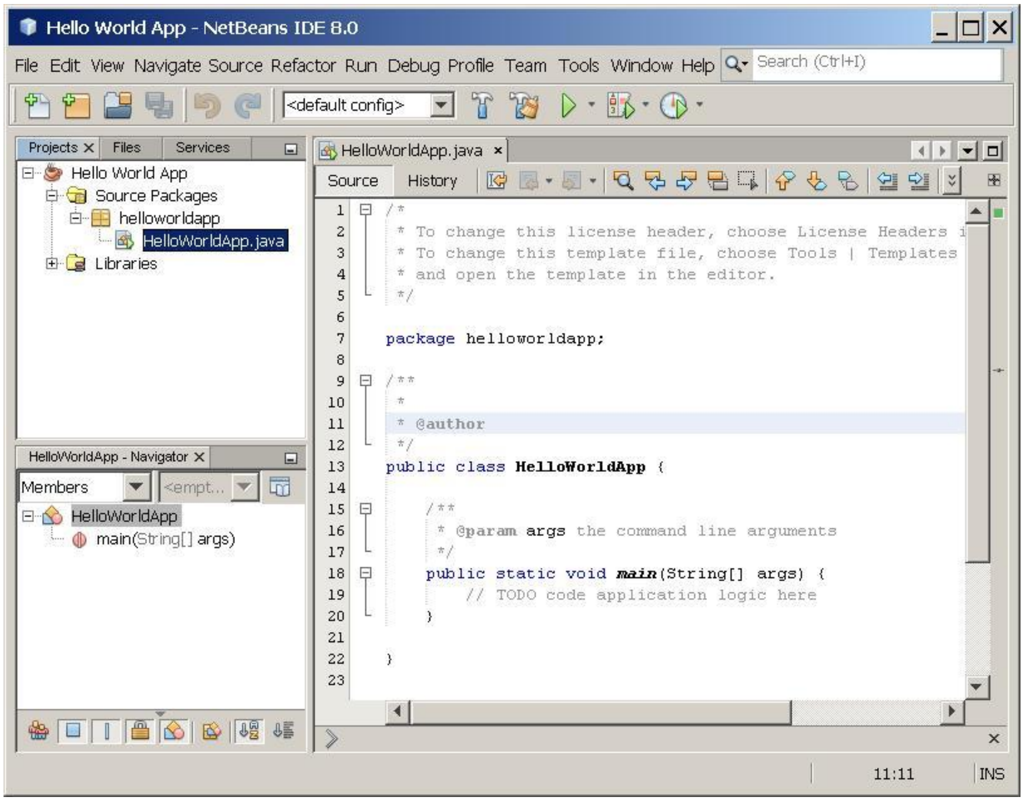
Different users and projects can share the same compilation libraries (see Help for details).

☑ Create Main Class helloworldapp.HelloWorldApp

1. NetBeans IDE, New Project wizard, Name and Location page.
2. Click Finish

The project is created and opened in the IDE. You should see the following components:

- The **Projects** window, which contains a tree view of the components of the project, including source files, libraries that your code depends on, and so on.
- The **Source Editor** window with a file called HelloWorldApp.java open.
- The **Navigator** window, which you can use to quickly navigate between elements within the selected class.



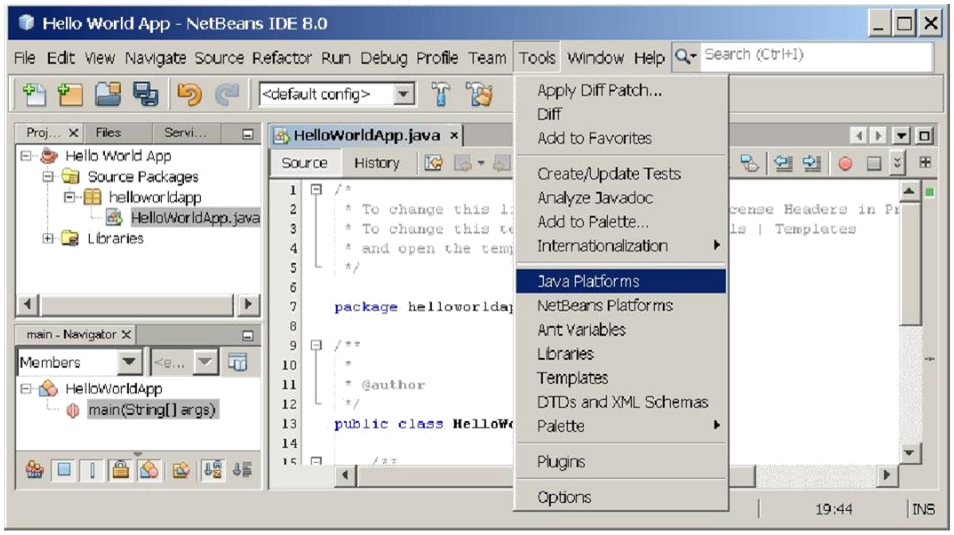
```
/*
 * To change this license header, choose License Headers i
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package helloworldapp;

/**
 *
 * @author
 */
public class HelloWorldApp {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

}

```

- NetBeans IDE with the HelloWorldApp project open.

Add JDK 8 to the Platform List (if necessary)

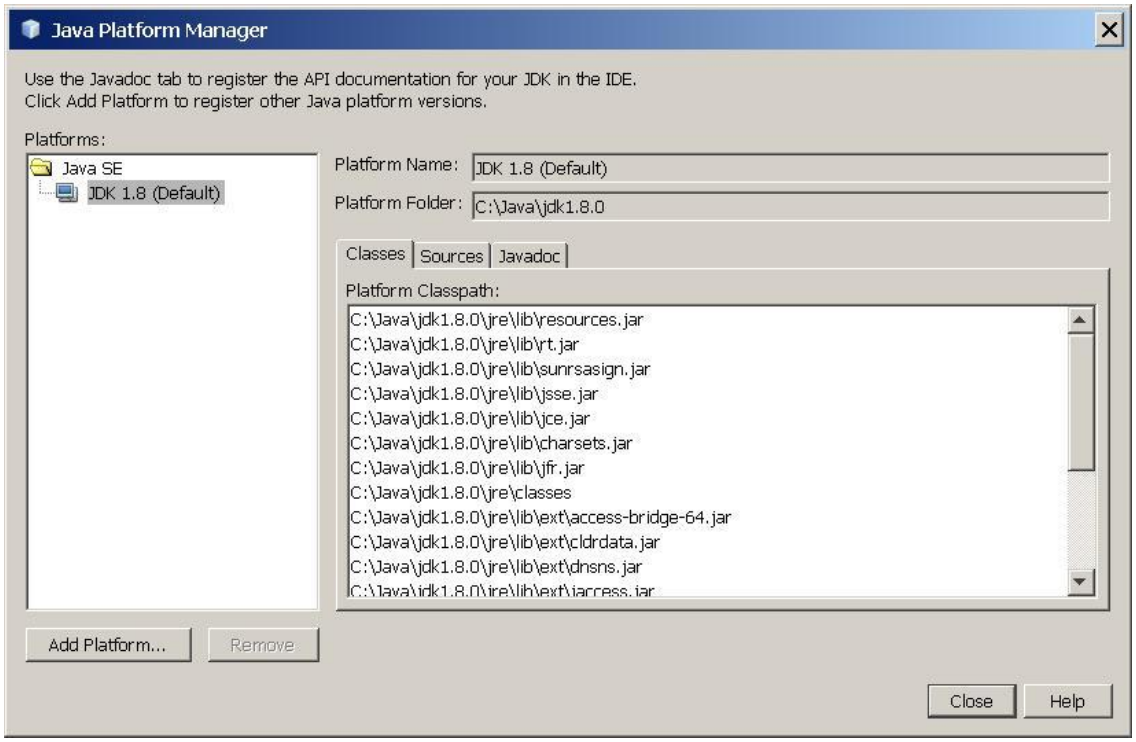
It may be necessary to add JDK 8 to the IDE's list of available platforms. To do this, choose Tools | Java Platforms as shown in the following figure

Selecting the Java Platform Manager from the Tools Menu



```
/*
 * To change this l
 * To change this t
 * and open the tem
 */

package helloworlda

/**
 *
 * @author
 */
public class HelloWo

    /**
```

If you don't see JDK 8 (which might appear as 1.8 or 1.8.0) in the list of installed platforms, click **Add Platform**, navigate to your JDK 8 install directory, and click **Finish**. You should now see this newly added platform:



Use the Javadoc tab to register the API documentation for your JDK in the IDE.
Click Add Platform to register other Java platform versions.

Platform Name: JDK 1.8 (Default)

Platform Folder: C:\Java\jdk1.8.0

Platform Classpath:

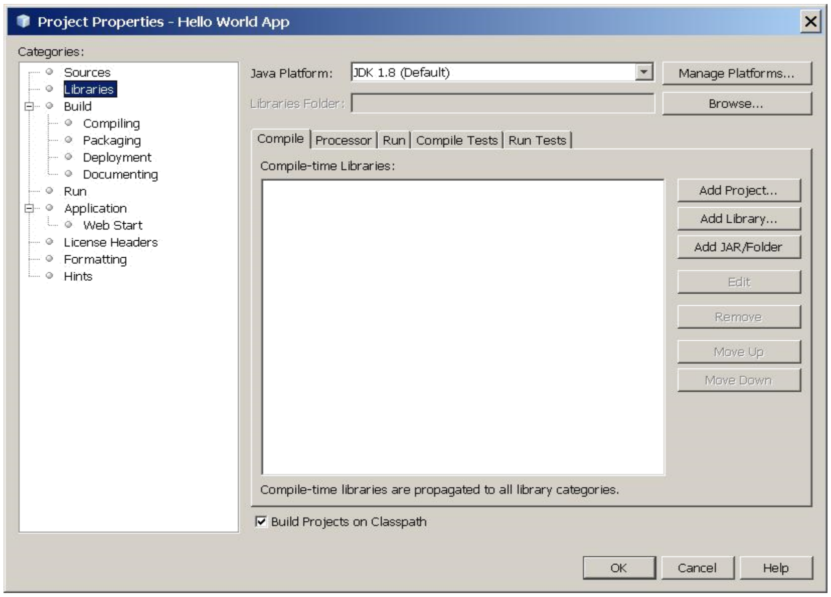
```
C:\Java\jdk1.8.0\jre\lib\resources.jar
C:\Java\jdk1.8.0\jre\lib\rt.jar
C:\Java\jdk1.8.0\jre\lib\sunrsasign.jar
C:\Java\jdk1.8.0\jre\lib\jsse.jar
C:\Java\jdk1.8.0\jre\lib\jce.jar
C:\Java\jdk1.8.0\jre\lib\charsets.jar
C:\Java\jdk1.8.0\jre\lib\jfr.jar
C:\Java\jdk1.8.0\jre\classes
C:\Java\jdk1.8.0\jre\lib\ext\access-bridge-64.jar
C:\Java\jdk1.8.0\jre\lib\ext\cldrdata.jar
C:\Java\jdk1.8.0\jre\lib\ext\dnsns.jar
C:\Java\jdk1.8.0\jre\lib\ext\jaccess.jar
```

The Java Platform Manager

To set this JDK as the default for all projects, you can run the IDE with the -- jdkhome switch on the command line, or by entering the path to the JDK in

the netbeans_j2sdkhome propertyofyour INSTALLATION_DIRECTORY/etc/netbeans.conf file.

To specify this JDK for the current project only, select **Hello World App** in the **Projects** pane, choose **File | Project Properties (Hello World App)**, click **Libraries**, then select **JDK 1.8** in the **Java Platform** pulldown menu. You should see a screen similar to the followig



Java Platform: JDK 1.8 (Default)

Compile-time libraries are propagated to all library categories.

☑ Build Projects on Classpath

The IDE is now configured for JDK 8.

Add Code to the Generated Source File

When you created this project, you left the **Create Main Class** checkbox selected in the **New Project** wizard. The IDE has therefore created a skeleton class for you. You can add the "Hello World!" message to the skeleton code by replacing the line:

```
// TODO code application logic herewith the line:
```

```
System.out.println("Hello World!"); // Display the string.
```

Optionally, you can replace these four lines of generated code:

```
/**
 *
 * @author
 */
```

with these lines:

```
/**
 * The HelloWorldApp class implements an application that
 * simply prints "Hello World!" to standard output.
 */
```

These four lines are a code comment and do not affect how the program runs. Later sections of this tutorial explain the use and format of code comments.

Be Careful When You Type

Note: Type all code, commands, and file names exactly as shown. Both the compiler(javac) and launcher (java) are*case-sensitive*, so you must capitalize consistently.

HelloWorldApp is *not* the same as helloworldapp.

Save your changes by choosing **File | Save**.

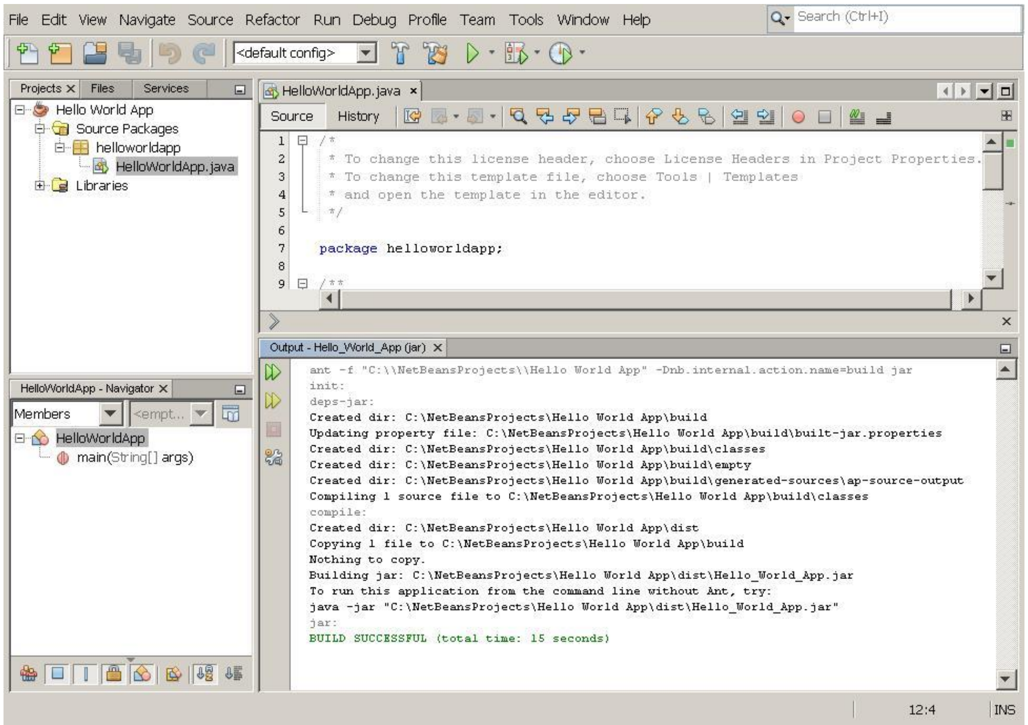
The file should look something like the following:

```
/*

 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */


package helloworldapp;


/**


 * The HelloWorldApp class implements an application that
 * simply prints "Hello World!" to standard output.
 */

public class HelloWorldApp {

    /**

     * @param args the command line
     arguments */

    public static void main(String[] args) {
        System.out.println("Hello World!"); // Display the string.

    }
```

Compile the Source File into a .class File

To compile your source file, choose **Run | Build Project (Hello World App)** from the IDE's main menu.

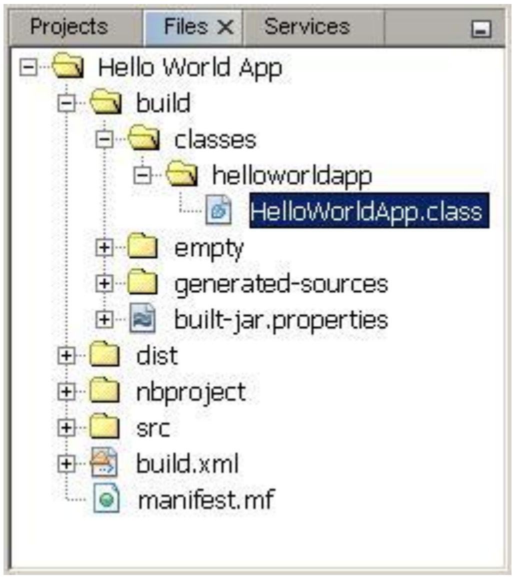
The Output window opens and displays output similar to what you see in the following figure:



```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package helloworldapp;

/**
```

```
ant -f "C:\\NetBeansProjects\\Hello World App" -Dnb.internal.action.name=build jar
init:
deps-jar:
Created dir: C:\NetBeansProjects\Hello World App\build
Updating property file: C:\NetBeansProjects\Hello World App\build\built-jar.properties
Created dir: C:\NetBeansProjects\Hello World App\build\classes
Created dir: C:\NetBeansProjects\Hello World App\build\empty
Created dir: C:\NetBeansProjects\Hello World App\build\generated-sources\ap-source-output
Compiling 1 source file to C:\NetBeansProjects\Hello World App\build\classes
compile:
Created dir: C:\NetBeansProjects\Hello World App\dist
Copying 1 file to C:\NetBeansProjects\Hello World App\build
Nothing to copy.
Building jar: C:\NetBeansProjects\Hello World App\dist\Hello_World_App.jar
To run this application from the command line without Ant, try:
java -jar "C:\NetBeansProjects\Hello World App\dist\Hello_World_App.jar"
jar:
BUILD SUCCESSFUL (total time: 15 seconds)
```

Output window showing results of building the HelloWorld project.



If the build output concludes with the statement BUILD SUCCESSFUL, congratulations!

You have successfully compiled your program!

If the build output concludes with the statement BUILD FAILED, you probably have a syntax error in your code. Errors are reported in the Output window as hyperlinked text. You double-click such a hyperlink to navigate to the source of an error. You can then fix the error and once again choose **Run | Build Project**.

When you build the project, the bytecode file HelloWorldApp.class is generated. You can see where the new file is generated by opening the **Files** window and expandingthe **Hello World App/build/classes/helloworldapp** node as shown in the following figure.

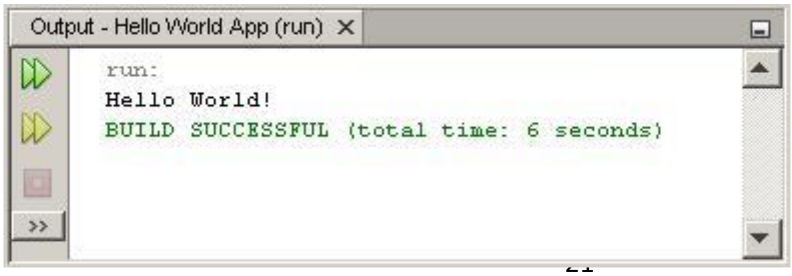
Files window, showing the generated .class file.

Now that you have built the project, you can run your program.

Run the Program

From the IDE's menu bar, choose **Run | Run Main Project**.

The next figure shows what you should now see.



```
run:
Hello World!
BUILD SUCCESSFUL (total time: 6 seconds)
```

The program prints "Hello World!" to the Output window (along with other output from the build script).

Congratulations! Your program works!

Continuing the Tutorial with the NetBeans IDE

The next few pages of the tutorial will explain the code in this simple application. After that, the lessons go deeper into core language features and provide many more examples.

Although the rest of the tutorial does not give specific instructions about using the NetBeans IDE, you can easily use the IDE to write and run the sample code. The following are some tips on using the IDE and explanations of some IDE behavior that you are likely to see:

- Once you have created a project in the IDE, you can add files to the project using the **New File** wizard. Choose **File | New File**, and then select a template in the wizard, such as the Empty Java File template.
- You can compile and run an individual file (as opposed to a whole project) using the IDE's **Compile File** (F9) and **Run File**(Shift-F6) commands. If you use the **Run Main Project** command, the IDE will run the file that the IDE associates as the main class of the main project. Therefore, if you create an additional class in your HelloWorldApp project and then try to run that file with the **Run MainProject** command, the IDE will run the HelloWorldApp file instead.
- You might want to create separate IDE projects for sample applications that include more than one source file.
- As you are typing in the IDE, a code completion box might periodically appear. You can either ignore the code completion box and keep typing, or you can select one of the suggested expressions. If you would prefer not to have the code

  completion box automatically appear, you can turn off the feature. Choose **Tools | Options | Editor**, click the **Code Completion** tab and clear the **Auto Popup Completion Window** checkbox.
- If you want to rename the node for a source file in the **Projects** window,

  choose **Refactor** from IDE's main menu. The IDE prompts you with the **Rename** dialog box to lead you through the options of renaming the class and the updating of code that refers to that class. Make the changes and click **Refactor** to apply the changes. This sequence of clicks might seem unnecessary if you have just a single class in your project, but it is very useful when your changes affect other parts of your code in larger projects.

```
}
```

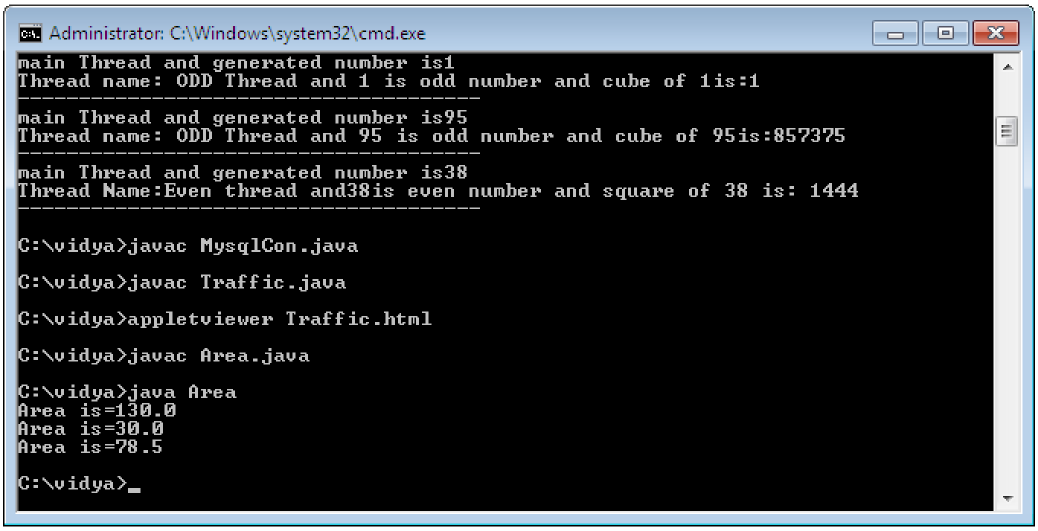
WEEK-2

2.Write a Java Program to create an abstract class named shape that contains two integers and an empty method named print Area(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the Class Shape. Each one of the classes contains only the method print Area() that prints the area of the given Shape.

```
abstract class Shape
{
double a,b;
abstract void printArea();
}

class Rectangle extends Shape
{
void printArea()
{
System.out.println("Area is="+(a*b));
}
}

class Triangle extends Shape
{
void printArea()
{
System.out.println("Area is="+(0.5*(a*b)));
}
}
```

```
class Circle extends Shape
{
void printArea()
{
System.out.println("Area is="+(a*b*b));
}
}
class Area
{
public static void main(String[] args)
{
Rectangle ob=new Rectangle();
ob.a=10;
ob.b=13;
ob.printArea();

Triangle ob1=new Triangle();
ob1.a=6;
ob1.b=10;
ob1.printArea();

Circle ob2=new Circle();
ob2.a=3.14;
ob2.b=5;
ob2.printArea();
}
```

OUTPUT:



```
main Thread and generated number is1
Thread name: ODD Thread and 1 is odd number and cube of 1is:1
--------------------------------------
main Thread and generated number is95
Thread name: ODD Thread and 95 is odd number and cube of 95is:857375
--------------------------------------
main Thread and generated number is38
Thread Name:Even thread and38is even number and square of 38 is: 1444
--------------------------------------

C:\vidya>javac MysqlCon.java

C:\vidya>javac Traffic.java

C:\vidya>appletviewer Traffic.html

C:\vidya>javac Area.java

C:\vidya>java Area
Area is=130.0
Area is=30.0
Area is=78.5

C:\vidya>_
```

WEEK-3

3. Write a Java Program that implements Bubble Sort algorithm for sorting in descending order and also show the number of interchanges occurred for the given set of integers.

```
public class BubbleSortExample {

    static void bubbleSort(int[] arr) {

        int n = arr.length;

        int temp = 0;

         for(int i=0; i < n; i++){

                 for(int j=1; j < (n-i); j++){

                          if(arr[j-1] > arr[j]){

                                 //swap elements

                                 temp = arr[j-1];
        arr[j-1] = arr[j];

        arr[j] = temp;

      }

       }

 }

 }

 public static void main(String[] args) {

    int arr[] ={3,60,35,2,45,320,5};


    System.out.println("Array Before Bubble Sort");

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

      System.out.print(arr[i] + " ");

    }

    System.out.println();


    bubbleSort(arr);//sorting array elements using bubble sort


    System.out.println("Array After Bubble Sort");

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

      System.out.print(arr[i] + " ");

    }


  }

}
```

Output:

```
Array Before Bubble Sort
```

```
3 60 35 2 45 320 5

Array After Bubble Sort

2 3 5 35 45 60 320
```

WEEK-4

4. Write a Java Program that implements Quick Sort algorithm for sorting a list of names in ascending order.

```
private static void quickSort(String[] a, int start, int end)

{

    // index for the "left-to-right scan"

    int i = start;

    // index for the "right-to-left scan"

    int j = end;


    // only examine arrays of 2 or more elements.

    if (j - i >= 1)

    {

        // The pivot point of the sort method is arbitrarily set to the first element int the
array.

        String pivot = a[i];

        // only scan between the two indexes, until they meet.

        while (j > i)

        {

            // from the left, if the current element is lexicographically less than the
(original)

            // first element in the String array, move on. Stop advancing the counter when
we reach
```

```
            // the right or an element that is lexicographically greater than the pivot String.
            while (a[i].compareTo(pivot) < 0 && i <= end && j > i){
                i++;
            }
            // from the right, if the current element is lexicographically greater than the (original)
            // first element in the String array, move on. Stop advancing the counter when we reach
            // the left or an element that is lexicographically less than the pivot String.
            while (a[j].compareTo(pivot) > 0 && j >= start && j >= i){
                j--;
            }
            // check the two elements in the center, the last comparison before the scans cross.
            if (j > i)
                swap(a, i, j);
        }
        // At this point, the two scans have crossed each other in the center of the array and stop.
        // The left partition and right partition contain the right groups of numbers but are not
        // sorted themselves. The following recursive code sorts the left and right partitions.

        // Swap the pivot point with the last element of the left partition.
        swap(a, start, j);
        // sort left partition
        quickSort(a, start, j - 1);
        // sort right partition
        quickSort(a, j + 1, end);

      }

    }

    /**

     * This method facilitates the quickSort method's need to swap two elements, Towers of Hanoi style.

     */

    private static void swap(String[] a, int i, int j)

    {

    String temp = a[i];

    a[i] = a[j];

    a[j] = temp;

    }
```

WEEK-5

5. Write a Java Program for the following:
Create a doubly linked list of elements.
Delete a given element from the above list
Display the contents of the list after deletion.import java.util.NoSuchElementException;

```
 public class DoublyLinkedListImpl<E> {



    private Node head;

    private Node tail;

    private int size;



    public DoublyLinkedListImpl() {

        size = 0;

    }
```

```
/**
 * this class keeps track of each element information
 * @author java2novice
 *
 */
private class Node {
    E element;
    Node next;
    Node prev;

    public Node(E element, Node next, Node prev) {
        this.element = element;
        this.next = next;
        this.prev = prev;
    }
}
/**
 * returns the size of the linked list
 * @return
 */
public int size() { return size; }

/**
 * return whether the list is empty or not
 * @return
 */
public boolean isEmpty() { return size == 0; }
```

```
/**
 * adds element at the starting of the linked list
 * @param element
 */
public void addFirst(E element) {
    Node tmp = new Node(element, head, null);
    if(head != null ) {head.prev = tmp;}
    head = tmp;
    if(tail == null) { tail = tmp;}
    size++;
    System.out.println("adding: "+element);
}

/**
 * adds element at the end of the linked list
 * @param element
 */
public void addLast(E element) {

    Node tmp = new Node(element, null, tail);
    if(tail != null) {tail.next = tmp;}
    tail = tmp;
    if(head == null) { head = tmp;}
    size++;
    System.out.println("adding: "+element);
}
```

```
/**
 * this method walks forward through the linked list
 */
public void iterateForward(){

    System.out.println("iterating forward..");
    Node tmp = head;
    while(tmp != null){
        System.out.println(tmp.element);
        tmp = tmp.next;
    }
}

/**
 * this method walks backward through the linked list
 */
public void iterateBackward(){

    System.out.println("iterating backword..");
    Node tmp = tail;
    while(tmp != null){
        System.out.println(tmp.element);
        tmp = tmp.prev;
    }
}
```

```
/**
 * this method removes element from the start of the linked list
 * @return
 */
public E removeFirst() {
    if (size == 0) throw new NoSuchElementException();
    Node tmp = head;
    head = head.next;
    head.prev = null;
    size--;
    System.out.println("deleted: "+tmp.element);
    return tmp.element;
}

/**
 * this method removes element from the end of the linked list
 * @return
 */
public E removeLast() {
    if (size == 0) throw new NoSuchElementException();
    Node tmp = tail;
    tail = tail.prev;
    tail.next = null;
    size--;
    System.out.println("deleted: "+tmp.element);
    return tmp.element;
}
```

```
    public static void main(String a[]){

        DoublyLinkedListImpl<Integer> dll = new DoublyLinkedListImpl<Integer>();
        dll.addFirst(10);
        dll.addFirst(34);
        dll.addLast(56);
        dll.addLast(364);
        dll.iterateForward();
        dll.removeFirst();
        dll.removeLast();
        dll.iterateBackward();
    }
}
```

Output:

```
adding: 10
adding: 34
adding: 56
adding: 364
iterating forward..
34
10
56
364
deleted: 34
deleted: 364
```

```
iterating backword..

56

10
```

WEEK-6

6. Write a Java Program to list all the files in a directory including files present in all its subdirectories.

```
public class ListDirectoryRecurisve {

   public void listDirectory(String dirPath, int level) {

      File dir = new File(dirPath);

      File[] firstLevelFiles = dir.listFiles();

      if (firstLevelFiles != null && firstLevelFiles.length > 0) {

         for (File aFile : firstLevelFiles) {

            for (int i = 0; i < level; i++) {

               System.out.print("\t");

            }

            if (aFile.isDirectory()) {

               System.out.println("[" + aFile.getName() + "]");

               listDirectory(aFile.getAbsolutePath(), level + 1);

            } else {

               System.out.println(aFile.getName());

            }

         }

      }

   }

   public static void main(String[] args) {
        ListDirectoryRecurisve test = new ListDirectoryRecurisve();
        String dirToList = System.getProperty("user.home") + File.separator +
"Documents";
        test.listDirectory(dirToList, 0);
    }
}
```

WEEK-7

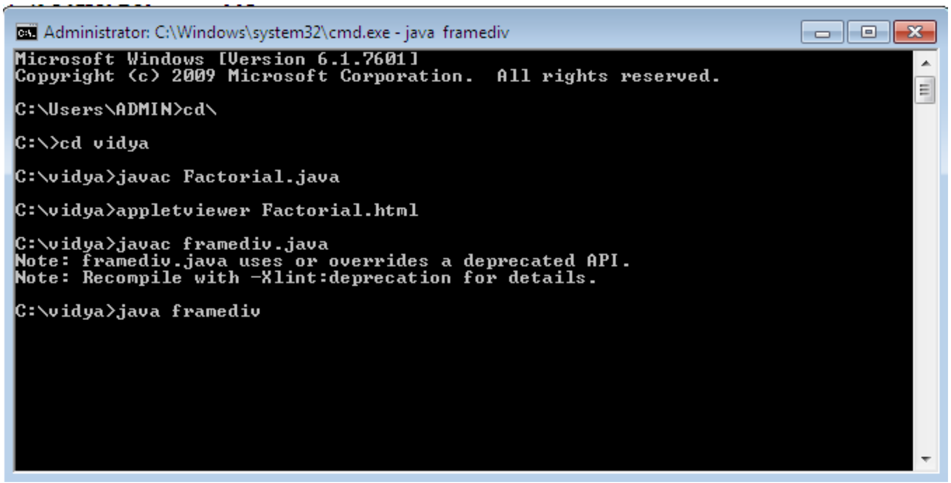
7.Write a Program that creates User Interface to perform Integer Divisons. The user enters two numbers in text fields, Num1 and Num2.The division of Num1 and Num2 is displayed in the result field when the divide button clicked. If Num1 or Num2 were not integer, the program would throw a Number Format Exception, If Num2 is Zero, and the program would throw an Arthematic Exception. Display the Exception in message dialog box.

```
import java.io.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
class framediv extends JFrame implements ActionListener
{
JPanel p;
JTextField t1,t2,t3;
JButton b;
JLabel msg;
public void actionPerformed(ActionEvent e)
{
```

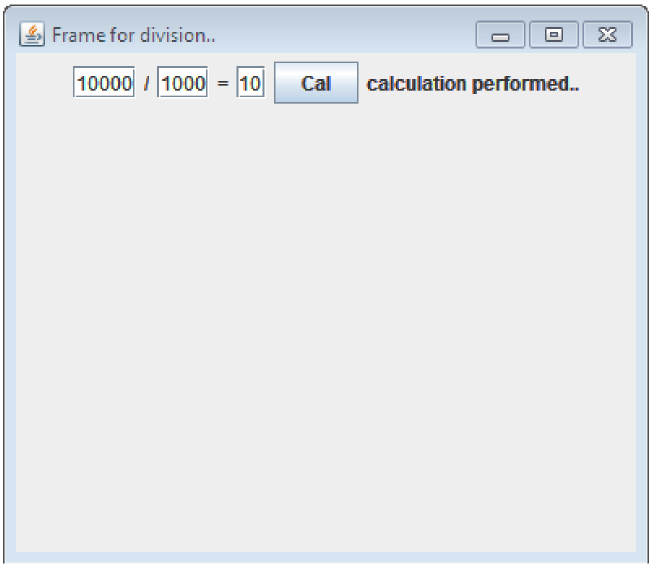
```
int n1,n2,n3;

JButton bt;

bt=(JButton)e.getSource();

if(bt.getLabel().equals("Cal"));

{

try

{

n1=Integer.parseInt(t1.getText());

n2=Integer.parseInt(t2.getText());

n3=n1/n2;

t3.setText(String.valueOf(n3));

msg.setText("calculation performed..");

}

catch(NumberFormatException ex)

{

msg.setText("Wrong Data Values are entered...");

}

catch(ArithmeticException ex)

{

msg.setText("Zero divide cannot possible");

}

}

}

public framediv()

{

setTitle("Frame for division..");

p=new JPanel();
```

```
t1=new JTextField("10000");
t2=new JTextField("1000");
t3=new JTextField("Result Here");
b=new JButton("Cal");
b.addActionListener(this);
p.add(t1);
p.add(new JLabel("/"));
p.add(t2);
p.add(new JLabel("="));
p.add(t3);
p.add(b);
msg=new JLabel("Click Cal Button to perform cal..");
p.add(msg);
add(p);
setSize(500,500);
setVisible(true);
}
public static void main(String args[])
{
new framediv();
}
}
```

OUTPUT:



```
Administrator: C:\Windows\system32\cmd.exe - java  framediv
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\ADMIN>cd\

C:\>cd vidya

C:\vidya>javac Factorial.java

C:\vidya>appletviewer Factorial.html

C:\vidya>javac framediv.java
Note: framediv.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\vidya>java framediv
```



Frame for division..

10000 / 1000 = 10 Cal calculation performed..

WEEK-8

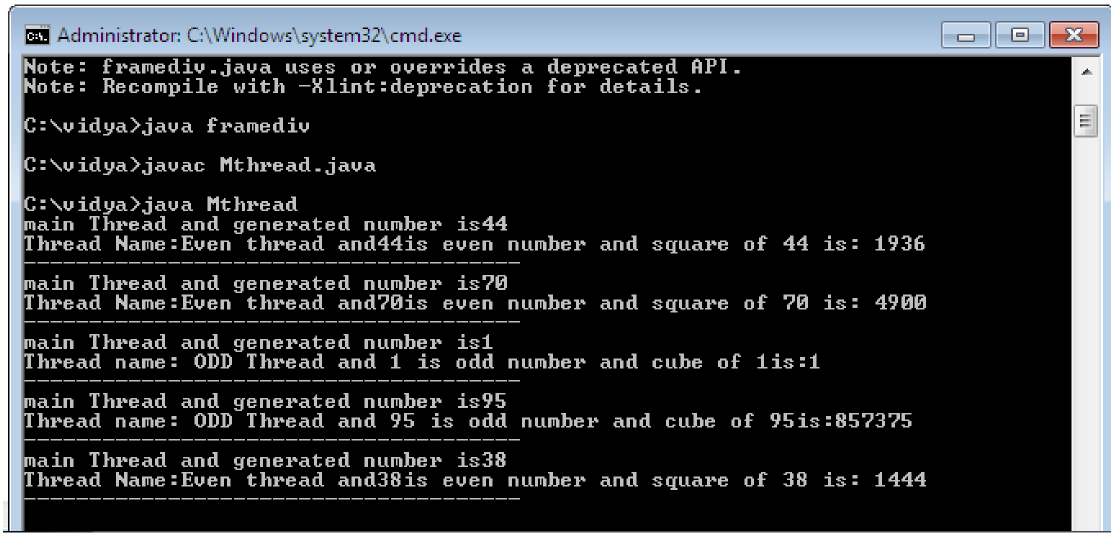
8.Write Java Program that implements a multithread application that has three threads. First thread generates random integer for every second and if the value is even, second thread computes the square of number and prints. If the value is odd, the third thread will print the value of cube of number.

```
import java.util.*;

class even implements Runnable

{

public int x;

public even(int x)

{

this.x=x;

}

public void run()

{

System.out.println("Thread Name:Even thread and"+x+"is even number and square of " + x
+" is: " + x * x);

}

}


class odd implements Runnable
```

```
{

public int x;

public odd(int x)

{

this.x=x;

}

public void run()

{

System.out.println("Thread name: ODD Thread and " + x +" is odd number and cube of " + x
+ "is:"+x * x * x);

}

}


class A extends Thread {

public String tname;

public Random r;

public Thread t1,t2;

public A(String s)

{

tname=s;

}

public void run(){

int num=0;

r= new Random();

try

{

for(int i=0; i<5;i++)
{
num=r.nextInt(100);
System.out.println("main Thread and generated number is"+num);
if(num%2==0)
{
t1=new Thread (new even(num));
t1.start();
}
else
{
t2=new Thread(new odd(num));
t2.start();
}
Thread.sleep(1000);
System.out.println("--------------------------------------");
}
}
catch(Exception ex)
{
System.out.println(ex.getMessage());
}
}
}

public class Mthread
{
public static void main(String[] agrs)
```

```
{
A a =new A("one");
a.start();
}
}
```

OUTPUT:



```
Note: framediv.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\vidya>java framediv

C:\vidya>javac Mthread.java

C:\vidya>java Mthread
main Thread and generated number is44
Thread Name:Even thread and44is even number and square of 44 is: 1936
--------------------------------------
main Thread and generated number is70
Thread Name:Even thread and70is even number and square of 70 is: 4900
--------------------------------------
main Thread and generated number is1
Thread name: ODD Thread and 1 is odd number and cube of 1is:1
--------------------------------------
main Thread and generated number is95
Thread name: ODD Thread and 95 is odd number and cube of 95is:857375
--------------------------------------
main Thread and generated number is38
Thread Name:Even thread and38is even number and square of 38 is: 1444
--------------------------------------
```

WEEK-9

9. Implement program on producer Consumer Problem by Using the concept of interthreadcommunication.

```
mport java.util.Vector;

import java.util.logging.Level;

import java.util.logging.Logger;


public class ProducerConsumerSolution {


   public static void main(String args[]) {
        Vector sharedQueue = new Vector();
        int size = 4;
        Thread prodThread = new Thread(new Producer(sharedQueue, size), "Producer");
        Thread consThread = new Thread(new Consumer(sharedQueue, size), "Consumer");
        prodThread.start();
        consThread.start();
    }
}

class Producer implements Runnable {

    private final Vector sharedQueue;
    private final int SIZE;

    public Producer(Vector sharedQueue, int size) {
        this.sharedQueue = sharedQueue;
        this.SIZE = size;
    }


    public void run() {
        for (int i = 0; i < 7; i++) {
            System.out.println("Produced: " + i);
            try {
                produce(i);
            } catch (InterruptedException ex) {
```

```
            Logger.getLogger(Producer.class.getName()).log(Level.SEVERE, null, ex);
        }

    }
  }

  private void produce(int i) throws InterruptedException {

    //wait if queue is full
    while (sharedQueue.size() == SIZE) {
      synchronized (sharedQueue) {
        System.out.println("Queue is full " + Thread.currentThread().getName()
                    + " is waiting , size: " + sharedQueue.size());

        sharedQueue.wait();
      }
    }

    //producing element and notify consumers
    synchronized (sharedQueue) {
      sharedQueue.add(i);
      sharedQueue.notifyAll();
    }
  }
}

class Consumer implements Runnable {
```

```
private final Vector sharedQueue;
private final int SIZE;

public Consumer(Vector sharedQueue, int size) {
    this.sharedQueue = sharedQueue;
    this.SIZE = size;
}

    public void run() {
    while (true) {
        try {
            System.out.println("Consumed: " + consume());
            Thread.sleep(50);
        } catch (InterruptedException ex) {
            Logger.getLogger(Consumer.class.getName()).log(Level.SEVERE, null, ex);
        }

    }
}

private int consume() throws InterruptedException {
    //wait if queue is empty
    while (sharedQueue.isEmpty()) {
        synchronized (sharedQueue) {
            System.out.println("Queue is empty " + Thread.currentThread().getName()
                                + " is waiting , size: " + sharedQueue.size());
```

```
                sharedQueue.wait();
            }
        }

        //Otherwise consume element and notify waiting producer
        synchronized (sharedQueue) {
            sharedQueue.notifyAll();
            return (Integer) sharedQueue.remove(0);
        }
    }
}
```

Output:

```
Produced: 0
Queue is empty Consumer is waiting , size: 0
Produced: 1
Consumed: 0
Produced: 2
Produced: 3
Produced: 4
Produced: 5
Queue is full Producer is waiting , size: 4
Consumed: 1
Produced: 6
Queue is full Producer is waiting , size: 4
Consumed: 2
```

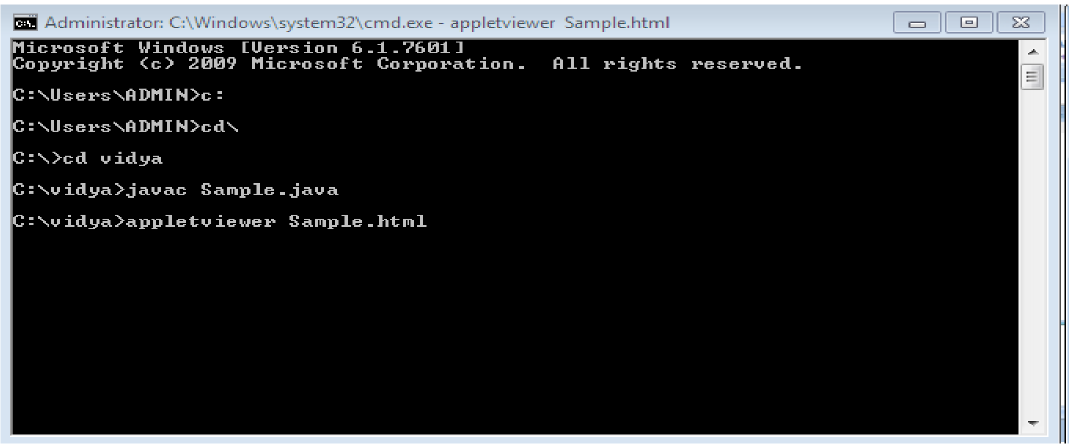
```
Consumed: 3
Consumed: 4
Consumed: 5
Consumed: 6
Queue is empty Consumer is waiting , size: 0
```

WEEK-10

a) **Develop an Applet in java that displays a Simple Message**

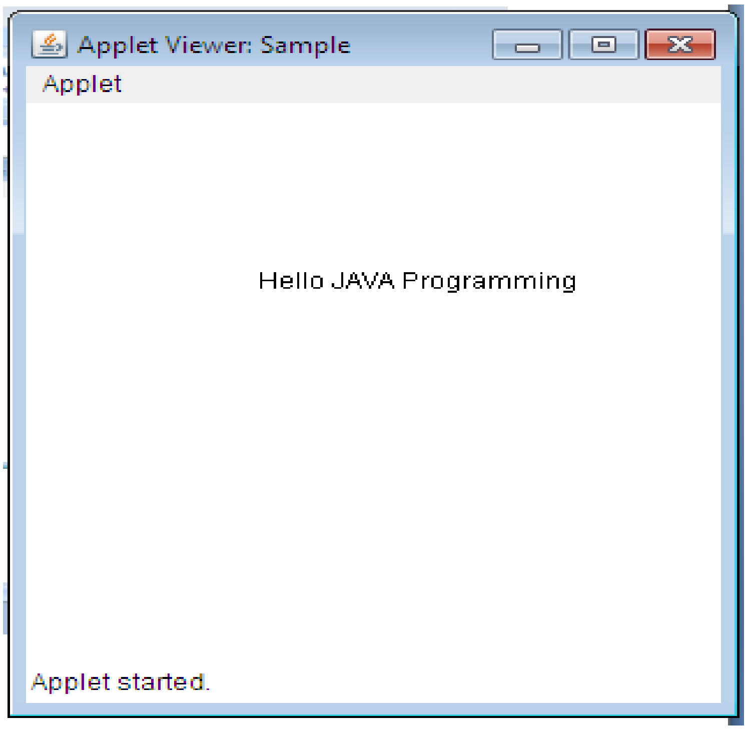
```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import javax.swing.*;
/*<applet code="Sample" height=300 width=300>
</applet>*/

public class Sample extends Applet
{
public void paint(Graphics g)
{
g.drawString("Hello JAVA Programming",100,100);
}
}
```

OUTPUT:



Administrator: C:\Windows\system32\cmd.exe - appletviewer Sample.html

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\ADMIN>c:

C:\Users\ADMIN>cd\

C:\>cd vidya

C:\vidya>javac Sample.java

C:\vidya>appletviewer Sample.html
```



Hello JAVA Programming

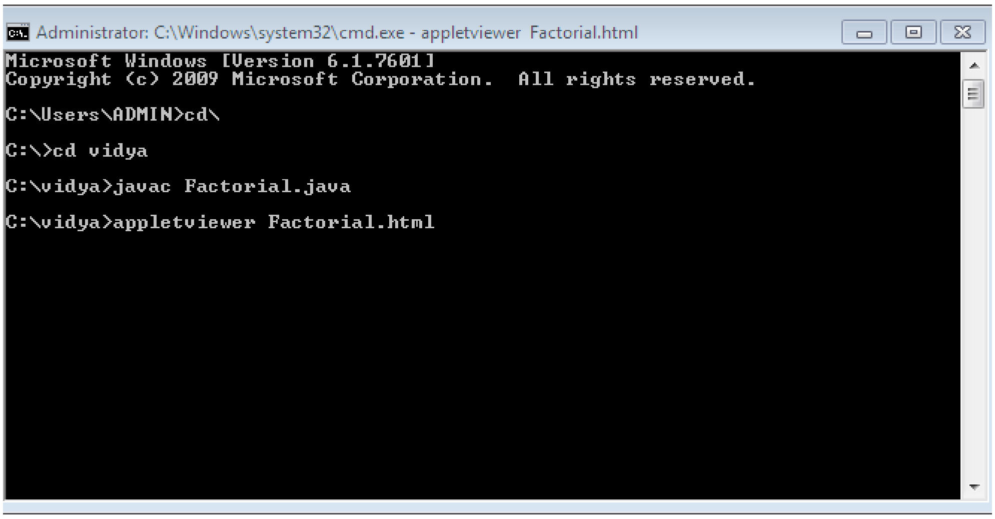
Applet started.

b) Develop an Applet in java that receives an integer in one TextField, and computes its Factorial value and returns it in another textfield, when button named “Compute” is clicked.

```
import java.awt.*;

import java.awt.event.*;

import java.applet.*;

import javax.swing.*;

/*<applet code="Factorial" height=300 width=300>

</applet>*/


public class Factorial extends JApplet implements ActionListener
```

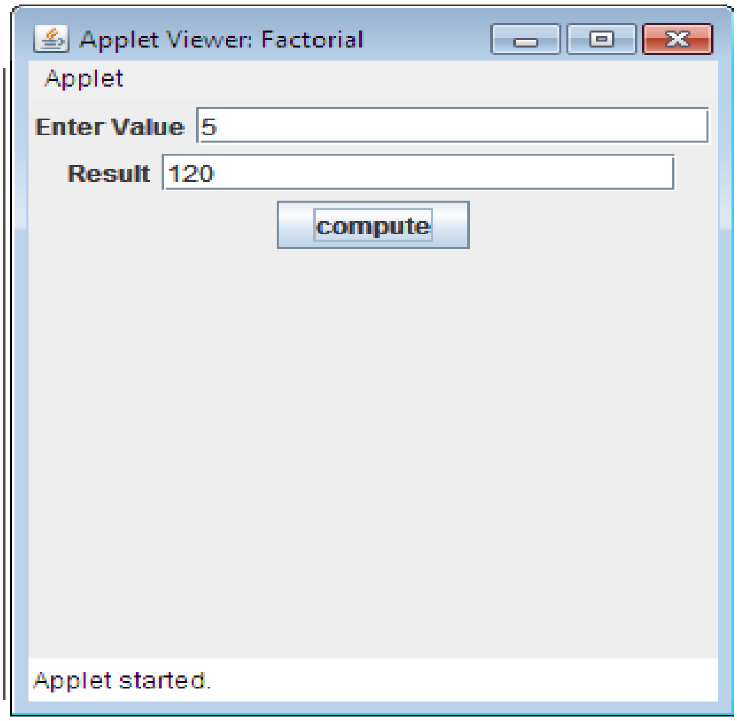
```
{

JLabel l1,l2;

JTextField t1,t2;

JButton b1;

public void init()
{
Container c=getContentPane();
c.setLayout(new FlowLayout());

l1=new JLabel("Enter Value");
l2=new JLabel("Result");

t1=new JTextField(20);
t2=new JTextField(20);

b1=new JButton("compute");

add(l1);
add(t1);
add(l2);
add(t2);

add(b1);
b1.addActionListener(this);
}
public void actionPerformed(ActionEvent ae)
{
if(ae.getActionCommand()=="compute")
{
int n=Integer.parseInt(t1.getText());
int f=1;
for(int i=1;i<=n;i++)
f=f*i;
t2.setText(String.valueOf(f));
}
}
}
```

OUTPUT:



Administrator: C:\Windows\system32\cmd.exe - appletviewer Factorial.html

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\ADMIN>cd\

C:\>cd vidya

C:\vidya>javac Factorial.java

C:\vidya>appletviewer Factorial.html
```



Enter Value 5

Result 120

Applet started.

WEEK-11

11.Write a Java Program that works as simple calculator .Use grid layout to arrange buttons for the digits and for the +,-,*,% operations .Add text field to display the results, Handle any possible exceptions like divide by zero. Suppose that a table named Table.txt is stored in atext file. The first line in the file is the header, and remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using Labels in GridLayout.

```
import java.awt.*;

import java.awt.event.*;

import java.applet.*;

import javax.swing.*;

/*<applet code="Calculator" height=300 width=300>

</applet>*/
```

```
public class Calculator extends JApplet implements ActionListener
{
JTextField t1;
JButton b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15,b16;

int a,b,c;
String op;

public void init()
{
Container c=getContentPane();
c.setLayout(new FlowLayout());
t1=new JTextField(20);
b1=new JButton("1");
b2=new JButton("2");
b3=new JButton("3");
b4=new JButton("4");
b5=new JButton("5");
b6=new JButton("6");
b7=new JButton("7");
b8=new JButton("8");
b9=new JButton("9");
b10=new JButton("0");
b11=new JButton("+");
b12=new JButton("-");
b13=new JButton("*");
b14=new JButton("/");
```

```
b15=new JButton("=");
b16=new JButton("C");
add(t1);
add(b1);
add(b2);
add(b3);
add(b4);
add(b5);
add(b6);
add(b7);
add(b8);
add(b9);
add(b10);
add(b11);
add(b12);
add(b13);
add(b14);
add(b15);
add(b16);
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);
b5.addActionListener(this);
b6.addActionListener(this);
b7.addActionListener(this);
b8.addActionListener(this);
```

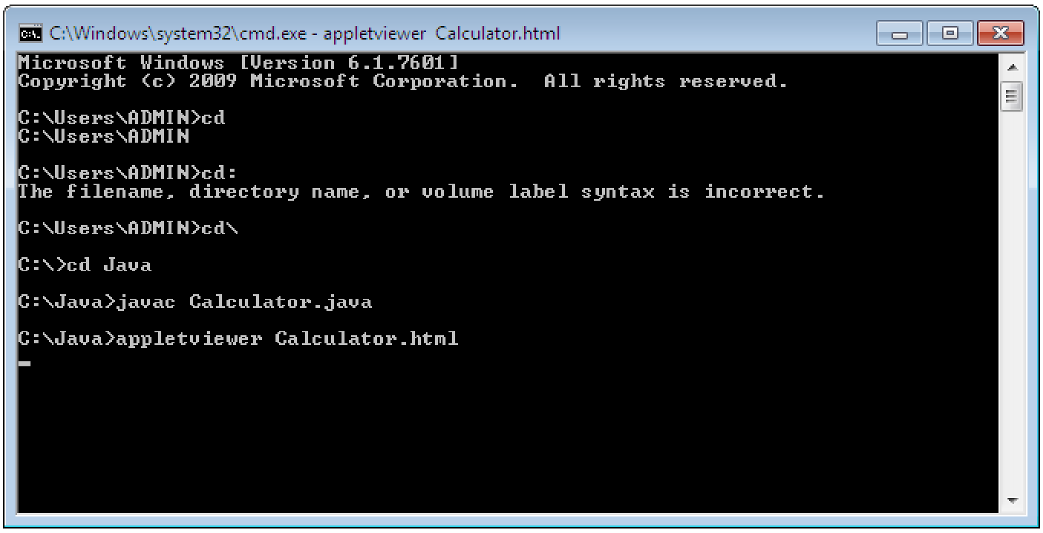
```
b9.addActionListener(this);

b10.addActionListener(this);

b11.addActionListener(this);

b12.addActionListener(this);

b13.addActionListener(this);

b14.addActionListener(this);

b15.addActionListener(this);

b16.addActionListener(this);

}


public void actionPerformed(ActionEvent ae)

{

if(ae.getActionCommand()=="1")

t1.setText(t1.getText()+"1");


if(ae.getActionCommand()=="2")

t1.setText(t1.getText()+"2");

if(ae.getActionCommand()=="3")

t1.setText(t1.getText()+"3");


if(ae.getActionCommand()=="4")

t1.setText(t1.getText()+"4");


if(ae.getActionCommand()=="5")

t1.setText(t1.getText()+"5");


if(ae.getActionCommand()=="6")
```

```
t1.setText(t1.getText()+"6");

if(ae.getActionCommand()=="7")
t1.setText(t1.getText()+"7");

if(ae.getActionCommand()=="8")
t1.setText(t1.getText()+"8");

if(ae.getActionCommand()=="9")
t1.setText(t1.getText()+"9");

if(ae.getActionCommand()=="0")
t1.setText(t1.getText()+"0");

if(ae.getActionCommand()=="+")
{
 a=Integer.parseInt(t1.getText());
 t1.setText("");
op="+";

}

if(ae.getActionCommand()=="-")
{
 a=Integer.parseInt(t1.getText());
 t1.setText("");
op="-";
```

```
}


if(ae.getActionCommand()=="*")
{
 a=Integer.parseInt(t1.getText());
 t1.setText("");
op="*";
}
if(ae.getActionCommand()=="/")
{
 a=Integer.parseInt(t1.getText());
 t1.setText("");
op="/";
}
if(ae.getActionCommand()=="=")
{


b=Integer.parseInt(t1.getText());
if(op=="+")
 c=a+b;


if(op=="-")
 c=a-b;


 if(op=="*")
 c=a*b;
```

```
if(op=="/")
 c=a/b;


 t1.setText(String.valueOf(c));
}
if(ae.getActionCommand()=="C")
{
t1.setText("");
}
}
}
```

OUTPUT:



C:\Windows\system32\cmd.exe - appletviewer Calculator.html

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\ADMIN>cd
C:\Users\ADMIN

C:\Users\ADMIN>cd:
The filename, directory name, or volume label syntax is incorrect.

C:\Users\ADMIN>cd\

C:\>cd Java

C:\Java>javac Calculator.java

C:\Java>appletviewer Calculator.html
_
```

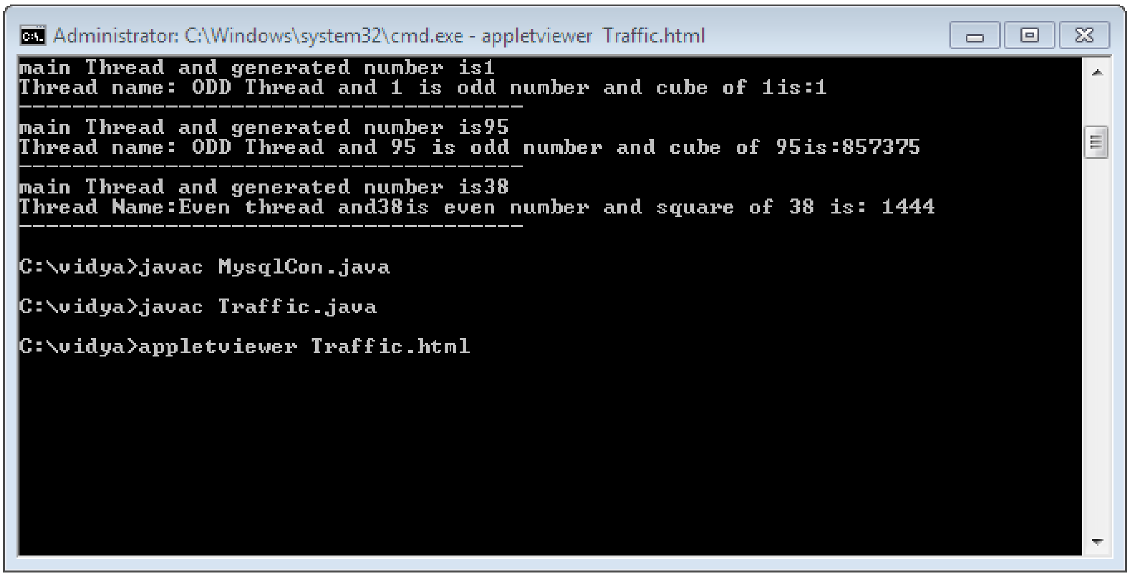
WEEK-12

12.Write a Java Program that simulates a Traffic Light. The program lets the user select one of three lights::red, yellow or Green with radio buttons. On selecting radio button, an appropriate message with “stop” or “Ready” or “GO” should appear above the buttons in selected color. Initially there is no message shown.

```
import java.awt.*;

import java.awt.event.*;

import java.applet.*;

import javax.swing.*;

/*<applet code="Traffic" height=300 width=300></applet>*/

public class Traffic extends JApplet implements ActionListener

{

JTextField t1;

JRadioButton b1,b2,b3;


public void init()

{

Container c=getContentPane();

c.setLayout(new FlowLayout());

t1=new JTextField(25);

b1=new JRadioButton("RED");

b2=new JRadioButton("YELLOW");

b3=new JRadioButton("GREEN");

add(t1);

add(b1);

add(b2);
add(b3);
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
ButtonGroup bg=new ButtonGroup();
bg.add(b1);
bg.add(b2);
bg.add(b3);
}
public void actionPerformed(ActionEvent ae)
{
if(ae.getActionCommand()=="RED")
{
t1.setText("STOP");
t1.setBackground(Color.RED);
}
if(ae.getActionCommand()=="YELLOW")
{
t1.setText("READY");
t1.setBackground(Color.YELLOW);
}
if(ae.getActionCommand()=="GREEN")
{
t1.setText("GO");
t1.setBackground(Color.GREEN);
}
}
```

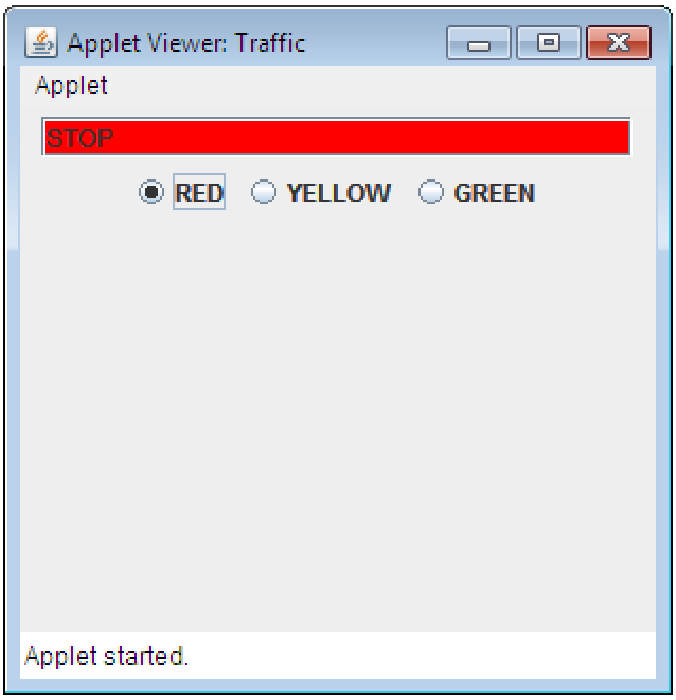
}

OUTPUT:



Administrator: C:\Windows\system32\cmd.exe - appletviewer Traffic.html

```
main Thread and generated number is1
Thread name: ODD Thread and 1 is odd number and cube of 1is:1
--------------------------------------
main Thread and generated number is95
Thread name: ODD Thread and 95 is odd number and cube of 95is:857375
--------------------------------------
main Thread and generated number is38
Thread Name:Even thread and38is even number and square of 38 is: 1444
--------------------------------------

C:\vidya>javac MysqlCon.java

C:\vidya>javac Traffic.java

C:\vidya>appletviewer Traffic.html
```



Applet Viewer: Traffic

Applet

STOP

● RED ○ YELLOW ○ GREEN

Applet started.

WEEK-13

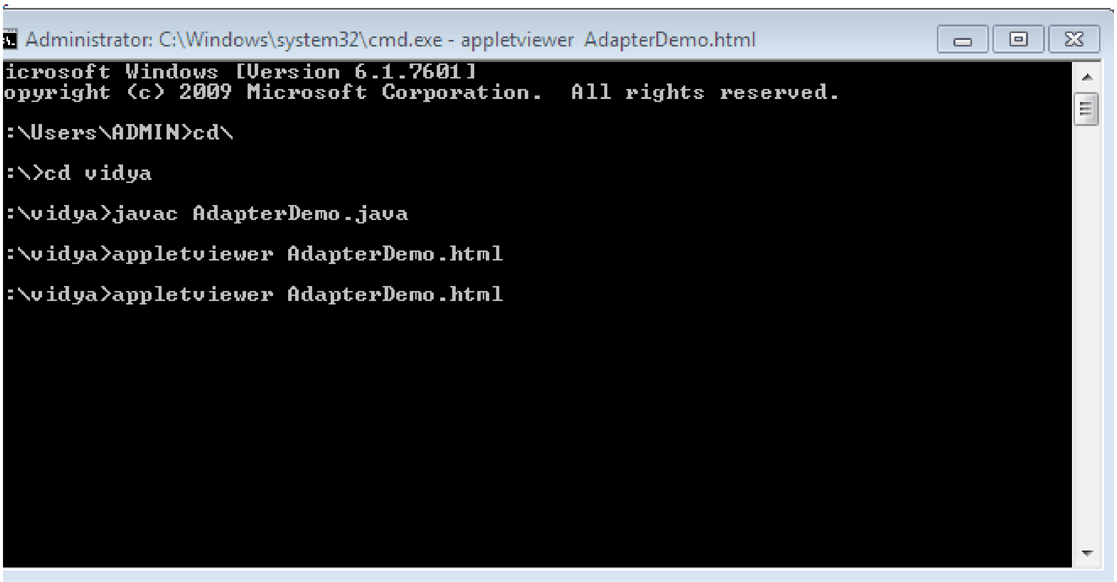
13.Write a Java Program that handles all mouse events and shows the event name at the center of the window when the mouse event is fired.(Use Adapter Classes)

```
import java.awt.*;

import java.applet.*;

import java.awt.event.*;


/*<applet code="AdapterDemo" width=300 height=100></applet>*/


public class AdapterDemo extends Applet

{

String msg="";

int mousex=0,mousey=0;


public AdapterDemo()

{

addMouseListener(new MyMouseAdapter(this));

addMouseMotionListener(new MyMouseMotionAdapter(this));

}

public void paint(Graphics g)

{

g.drawString(msg,mousex,mousey);

}

}


class MyMouseAdapter extends MouseAdapter

{
```

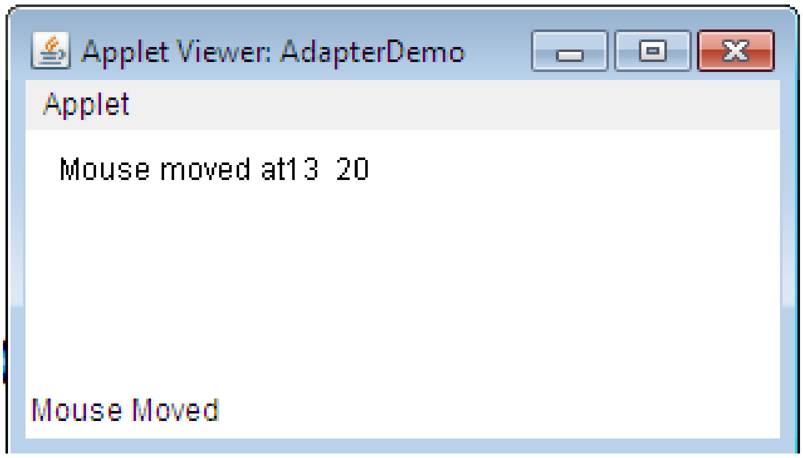
```
AdapterDemo ad;
public MyMouseAdapter(AdapterDemo ad)
{
this.ad=ad;
}
public void mouseExited(MouseEvent me)
{
ad.mousex=me.getX();
ad.mousey=me.getY();
ad.msg="Mouse Exited at"+ad.mousex+"  "+ad.mousey;
ad.repaint();
ad.showStatus("Mouse Exited");
}
public void mouseEntered(MouseEvent me)
{
ad.mousex=me.getX();
ad.mousey=me.getY();
ad.msg="Mouse entered at "+ad.mousex+"  "+ad.mousey;
ad.repaint();
ad.showStatus("Mouse Entered");
}
public void mouseReleased(MouseEvent me)
{
ad.mousex=me.getX();
ad.mousey=me.getY();
ad.msg="Mouse Released at"+ad.mousex+"  "+ad.mousey;
ad.repaint();
ad.showStatus("Mouse Released");
}
public void mousePressed(MouseEvent me)
{
ad.mousex=me.getX();
ad.mousey=me.getY();
ad.msg="Mouse Pressed at"+ad.mousex+"  "+ad.mousey;
ad.repaint();
ad.showStatus("Mouse Pressed");
}
public void mouseClicked(MouseEvent me)
{
ad.mousex=me.getX();
ad.mousey=me.getY();
ad.msg="Mouse Clicked at"+ad.mousex+"  "+ad.mousey;
ad.repaint();
ad.showStatus("Mouse Clicked");
}
}

class MyMouseMotionAdapter extends MouseMotionAdapter
{
AdapterDemo ad;

public MyMouseMotionAdapter(AdapterDemo ad)
{
this.ad=ad;
```

```
}
public void mouseMoved(MouseEvent me)
{
ad.mousex=me.getX();
ad.mousey=me.getY();
ad.msg="Mouse moved at"+ad.mousex+"  "+ad.mousey;
ad.repaint();
ad.showStatus("Mouse Moved");
}
public void mouseDragged(MouseEvent me)
{
ad.mousex=me.getX();
ad.mousey=me.getY();
ad.msg="Mouse Dragged at"+ad.mousex+"  "+ad.mousey;
ad.repaint();
ad.showStatus("Mouse Dragged");
}
}
```

OUTPUT:


Administrator: C:\Windows\system32\cmd.exe - appletviewer AdapterDemo.html

```
icrosoft Windows [Version 6.1.7601]
opyright (c) 2009 Microsoft Corporation.  All rights reserved.

:\Users\ADMIN>cd\

:\>cd vidya

:\vidya>javac AdapterDemo.java

:\vidya>appletviewer AdapterDemo.html

:\vidya>appletviewer AdapterDemo.html
```



```
Mouse moved at13  20
```

```
Mouse Moved
```

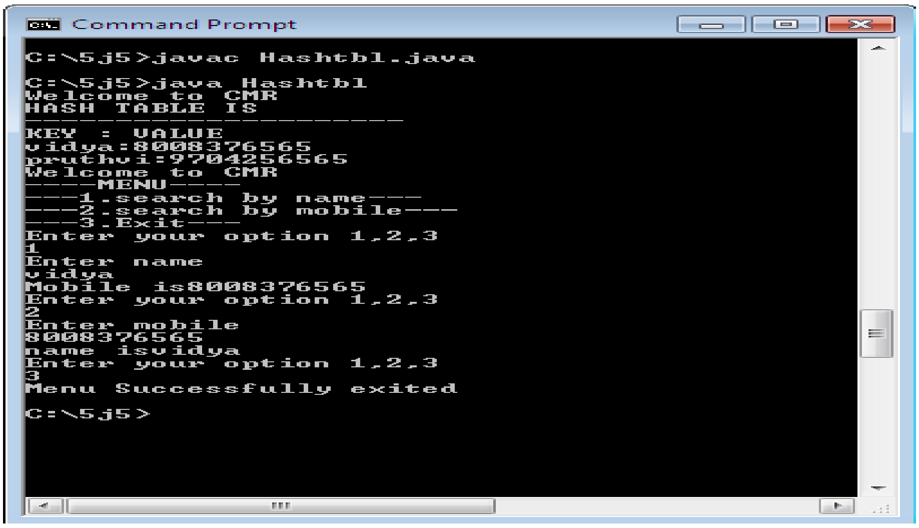
WEEK-14

14.Write a java program that loads names and phone numbers from the text file where data is organized as one line per record and each field in record are separated by a tab(\t).It takes a name or phone number as input and prints corresponding other value from hash table(hint: use Hash Table)

```
import java.util.*;
import java.io.*;
public class Hashtbl{
public static void main(String[] args){
try{
FileInputStream fs=new FileInputStream("E:\\phone.txt");
Scanner sc=new Scanner(fs).useDelimiter("\\s+");
Hashtable<String,String>ht=new Hashtable<String,String>();
String[] arrayList;
String a;
System.out.println("Welcome to CMR");
System.out.println("HASH TABLE IS");
System.out.println("--------------------");
System.out.println("KEY : VALUE");
while(sc.hasNext()){
a=sc.nextLine();
arrayList=a.split("\\s+");
ht.put(arrayList[0],arrayList[1]);
System.out.println(arrayList[0]+":"+arrayList[1]);
}
System.out.println("Welcome to CMR");
System.out.println("----MENU----");
System.out.println("---1.search by name---");
System.out.println("---2.search by mobile---");
System.out.println("---3.Exit---");
String opt="";
String name,mobile;
```

```
Scanner s=new Scanner(System.in);

while(opt!="3"){

System.out.println("Enter your option 1,2,3");

opt=s.next();

switch(opt){

case "1":{

System.out.println("Enter name");

name=s.next();

if(ht.containsKey(name)){

System.out.println("Mobile is"+ht.get(name));

}else{

System.out.println("not found");

}

}

break;

case "2":{

System.out.println("Enter mobile");

mobile=s.next();

if(ht.containsValue(mobile)){

for(Map.Entry e : ht.entrySet()){

if(mobile.equals(e.getValue())){

System.out.println("name is"+e.getKey());

}

}

}else{

System.out.println("not found");

}
}
break;
case "3":{
opt="3";
System.out.println("Menu Successfully exited");
}
break;
default:
System.out.println("choose option between 1 and 3");
break;
}
}
}catch(Exception ex){
System.out.println(ex.getMessage());
}
}
}
```

OUTPUT:



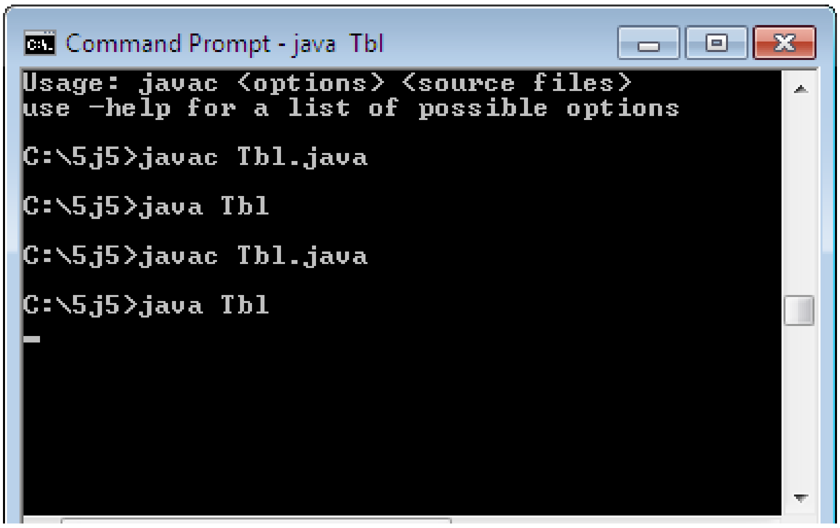
```
C:\5j5>javac Hashtbl.java

C:\5j5>java Hashtbl
Welcome to CMR
HASH TABLE IS
---------------------
KEY  :  VALUE
vidya:8008376565
pruthvi:9704256565
Welcome to CMR
----MENU----
----1.search by name---
----2.search by mobile---
----3.Exit---
Enter your option 1,2,3
1
Enter name
vidya
Mobile is8008376565
Enter your option 1,2,3
2
Enter mobile
8008376565
name isvidya
Enter your option 1,2,3
3
Menu Successfully exited

C:\5j5>
```

WEEK-15

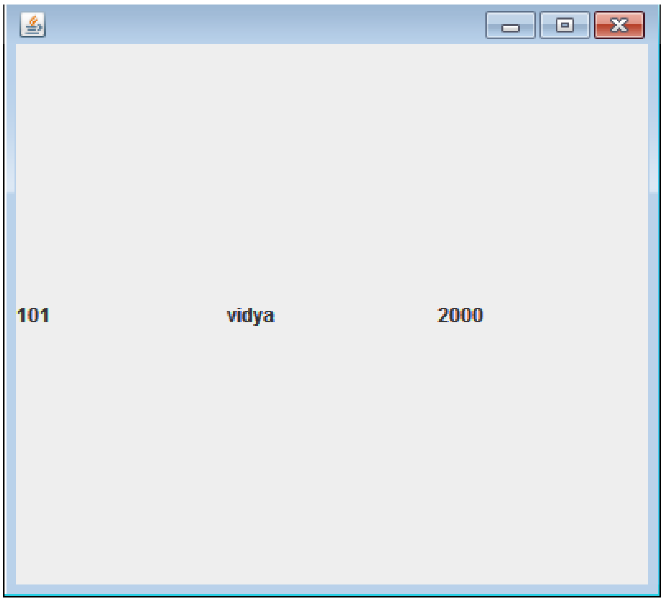
15.Suppose that a table named Table.txt is stored in a text file. The First line in the file is the header, and the remaining lines correspond to rows in table. The elements are separated by commas. Write java program to display the table using Labels in Grid Layout.

```
import java.io.*;

import java.util.*;

import java.awt.*;

import java.awt.event.*;

import javax.swing.*;

import javax.swing.event.*;


class A extends JFrame {

public A() {

setSize(400, 400);

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

GridLayout g = new GridLayout(0, 3);
}
setLayout(g);
try {
FileInputStream fin = new FileInputStream("E:\\student.txt");
Scanner sc = new Scanner(fin).useDelimiter(",");
String[] arrayList;
String a;
while (sc.hasNextLine()) {
a = sc.nextLine();
arrayList = a.split(",");
for (String i : arrayList) {
add(new
JLabel(i));
}
}
} catch (Exception ex) {
}

setVisible(true);
}
}
public class Tbl {
public static void main(String[] args) {
A a = new A();
}
}
```

OUTPUT:



```
Usage: javac <options> <source files>
use -help for a list of possible options

C:\5j5>javac Tbl.java

C:\5j5>java Tbl

C:\5j5>javac Tbl.java

C:\5j5>java Tbl
_
```



```
101                 vidya               2000
```