JAVA PROGRAMMING LAB

LAB MANUAL

Subject Code: CS408PC

Regulations : R18-JNTUH

Class : II Year B.Tech. CSE II Semester

Prepared by

Mr Romy SinhaAssistant Professor, CSE

Department of Computer Science and Engineering

BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY

Ibrahimpatnam - 501 510, Hyderabad

VISION AND MISSION OF THE INSTITUTION

Vision

To achieve the autonomous & university status and spread universal education by inculcating discipline, character and knowledge into the young minds and mould them into enlightened citizens.

Mission

Our mission is to impart education, in a conducive ambience, as comprehensive as possible, with the support of all the modern technologies and make the students acquire the ability and passion to work wisely, creatively and effectively for the betterment of our society.

VISION AND MISSION OF CSE/IT DEPARTMENT

Vision

Serving the high quality educational needs of local and rural students within the core areas of Computer Science and Engineering and Information Technology through a rigorous curriculum of theory, research and collaboration with other disciplines that is distinguished by its impact on academia, industry and society.

Mission

The Mission of the department of Computer Science and Engineering is

- To work closely with industry and research organizations to provide high quality computer education in both the theoretical and applications of Computer Science and Engineering.
- The department encourages original thinking, fosters research and development, evolve innovative applications of technology.

BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY

Ibrahimpatnam - 501 510, Hyderabad

COMPUTER SCIENCE AND ENGINEERING

Program Educational Objectives (PEOs):

A graduate of the Computer Science and Engineering Program should:

Program Educational Objective 1: (PEO1)
The graduates of Computer Science and Engineering will have successful career in technology or managerial functions.

Program Educational Objective 2: (PEO2)
The graduates of the program will have solid technical and professional foundation to continue higher studies.

Program Educational Objective 3: (PEO3)
The graduates of the program will have skills to develop products, offer services and create new knowledge.

Program Educational Objective 4: (PEO4)
The graduates of the program will have fundamental awareness of Industry processes, tools and technologies.

Program Outcomes (POs):

PO1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	**Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	**Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	**Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assess Societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7	**Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	**Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	**Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	**Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs):

PSO1	**Software Development and Research Ability:** Ability to understand the structure and development methodologies of software systems. Possess professional skills and knowledge of software design process. Familiarity and practical competence with a broad range of programming language and open source platforms. Use knowledge in various domains to identify research gaps and hence to provide solution to new ideas and innovations.
PSO2	**Foundation of mathematical concepts:** Ability to apply the acquired knowledge of basic skills, principles of computing, mathematical foundations, algorithmic principles, modeling and design of computer- based systems in solving real world engineering Problems.
PSO3	**Successful Career:** Ability to update knowledge continuously in the tools like Rational Rose, MATLAB, Argo UML, R Language and technologies like Storage, Computing, Communication to meet the industry requirements in creating innovative career paths for immediate employment and for higher studies.

Course Objectives:

- To write programs using abstract classes.
- To write programs for solving real world problems using java collection frame work.
- To write multithreaded programs.
- To write GUI programs using swing controls in Java.
- To introduce java compiler and eclipse platform.

Course Outcomes:

- Able to write programs for solving real world problems using java collection frame work.
- Able to write programs using abstract classes.
- Able to write multithreaded programs.
- Able to write GUI programs using swing controls in Java

INDEX	
S. No.	**List of Experiments**
1	Use Eclipse or Netbean 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.
2	Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result. Handle any possible exceptions like divided by zero.
3	Develop an applet that displays a simple message. Develop an applet that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named "Compute" is clicked
4	Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an ArithmeticException. Display the exception in a message dialog box.
5	Write a Java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number
6	Write a java program for the following: i)create a doubly linked list of elements ii)Delete a given elements from above list iii)Display the contents of list after deletion
7	Write a Java Program that simulates a Traffic Light. The program lets the use select one of three lights :red, yellow Green with radiobuttons.On selecting radio button, an appropriate message with "stop" or "Ready" or "GO" should appear above the button in selected color.Intially ,there is no message shown
8	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

9	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 the table. The elements are separated by commas. Write a java program to display the table using Labels in Grid Layout
10	Write a Java program that handles all mouse events and shows the event name at the center of the window when a mouse event is fired (Use Adapter classes)
11	Write a Java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab (\t). It takes a name or phone number as input and prints the corresponding other value from the hash table (hint: use hash tables)
12	Write a java program that correctly implements the producer-consumer problem using the concept of inter thread communication
13	Write a java program to list all the files in a directory including files
14	Write a Java program that implements Quick sort algorithm for sorting a list of names in ascending order
15	Write a Java program that implements Bubble sort algorithm for sorting in descending order and also shows the number of interchanges occurred for the given set of integers.

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 (javacompiler)
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.

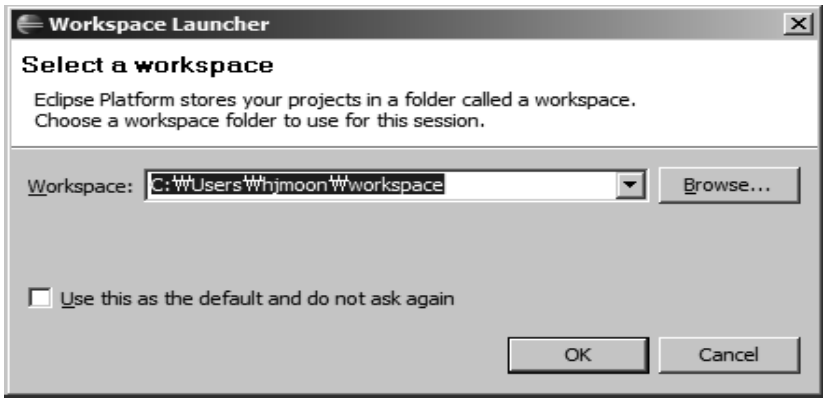
$CLASSPATH= C:\Program Files\Java\Jre\bin;

WEEK-1

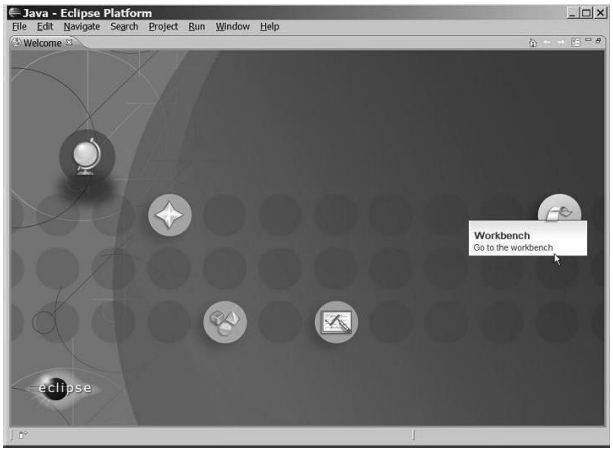
1) Use Eclipse or Net bean platform and acquaint with the various menus. Create a test project, add a test class, and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods, and classes. Try debug step by step with a small program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

Aim: To create and test java projects in Eclipse and Netbean platform

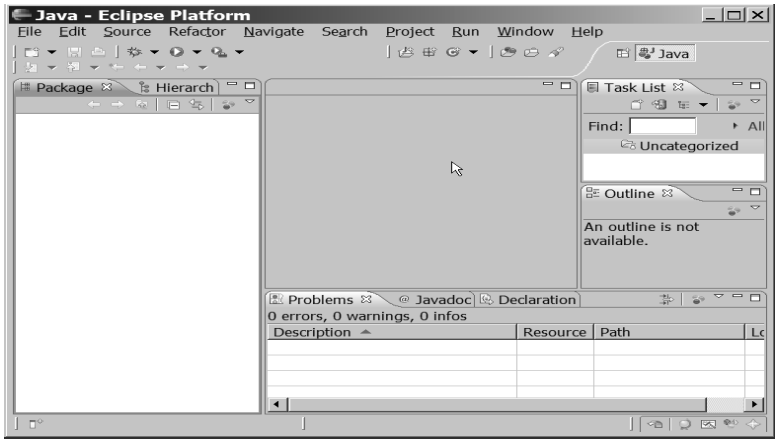
Initializing Eclipse : When you develop Java applications in Eclipse, it stores all the created files in a directory called "workspace". When Eclipse is run for the first time, it will ask you where you want the workspace to be placed:



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, you
Will see the following welcome window:

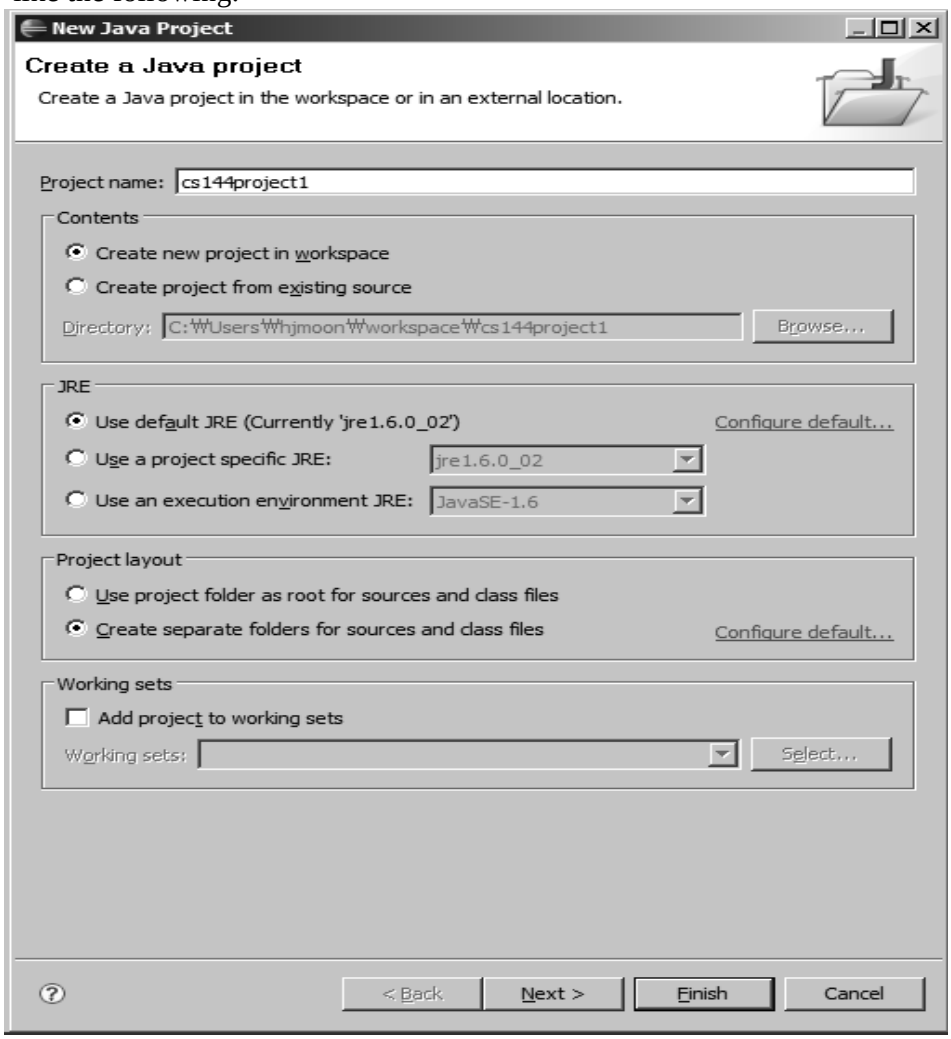


Click the "Workbench" icon on the right, which will lead you to the main Eclipse window:

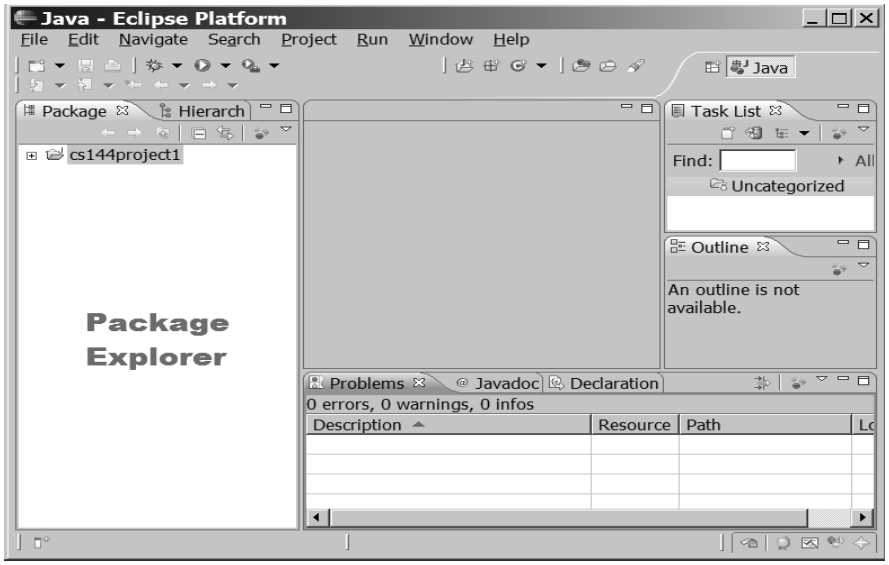


Creating a Project

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:

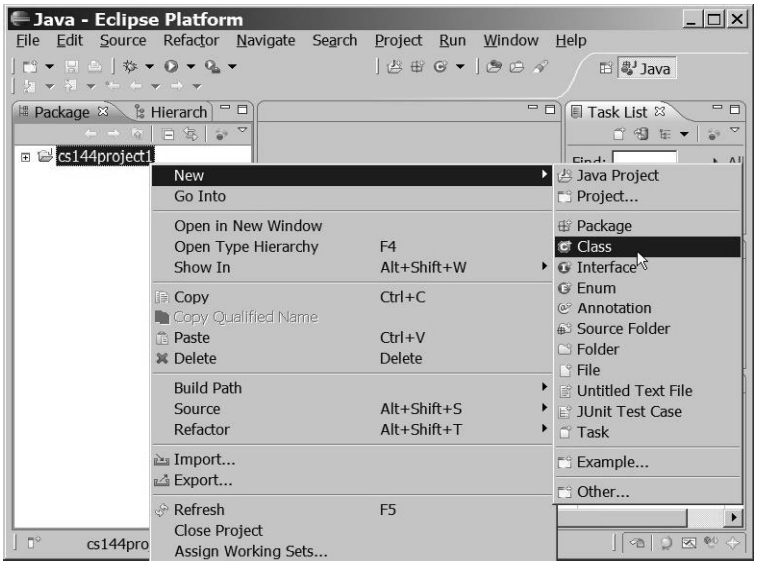


Type your project name (say, cs144project1) in the "Project name" field and click Finish. Then the name of your newly created project will appear on the left side of the Eclipse window (this part of the window is called "Package explorer pane"):

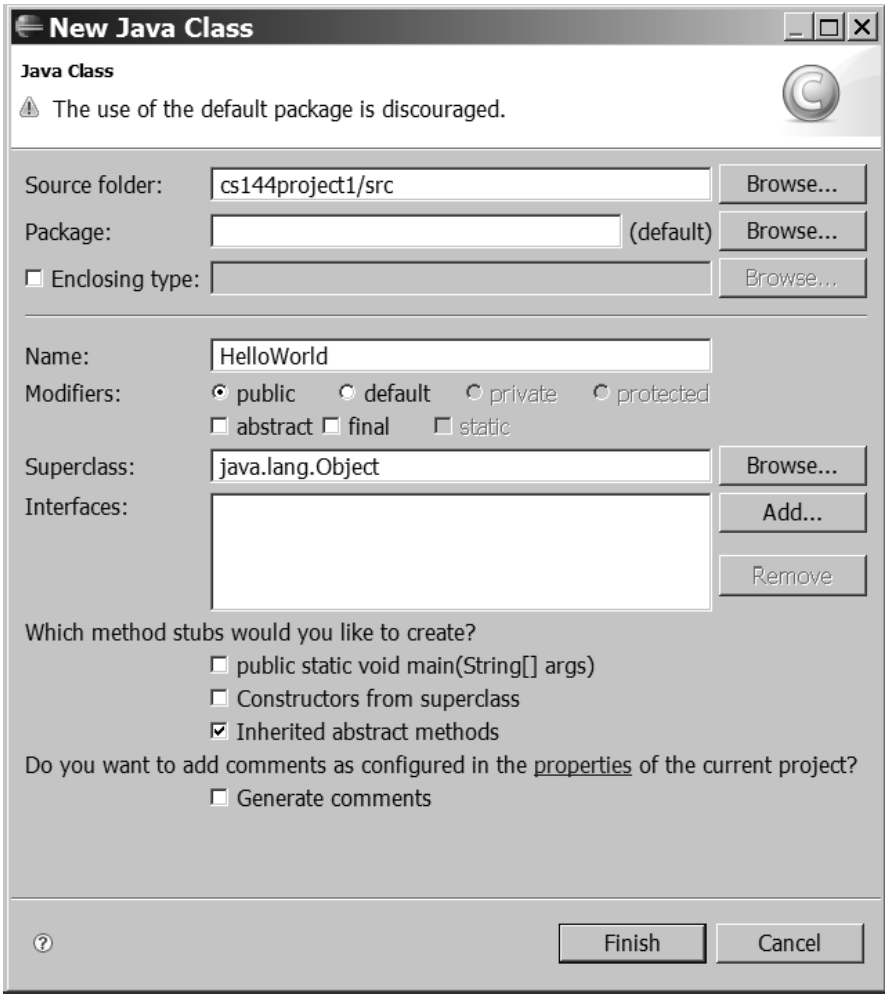


As you create more projects in Eclipse, other project names will appear in the Package explorer pane and you will be able to switch between your projects by clicking the name of a project.

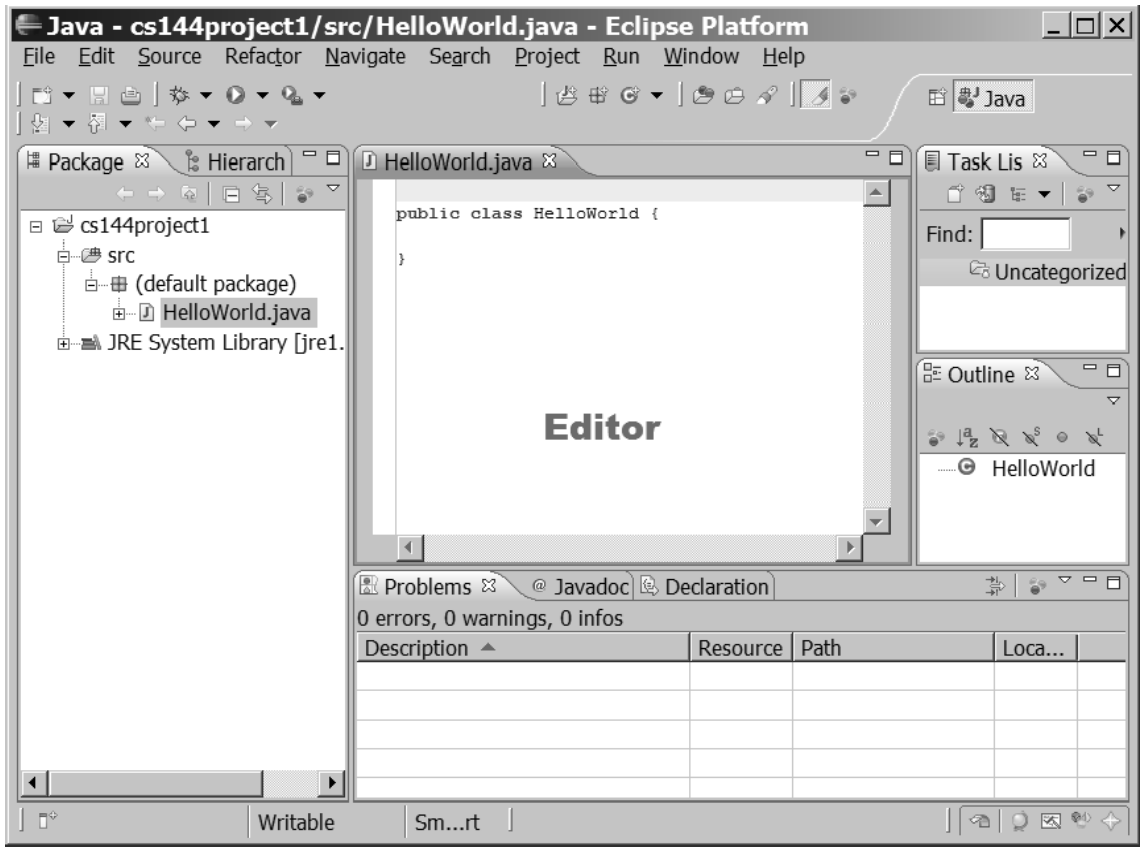
Adding a New File to a Project : Now that you've created your first project, you now want to create a new Java file (with .java extension) and add it into your project. To create a new Java file, right click on the name of your project (cs144project1) in the Package explorer pane and select New -> Class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



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



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

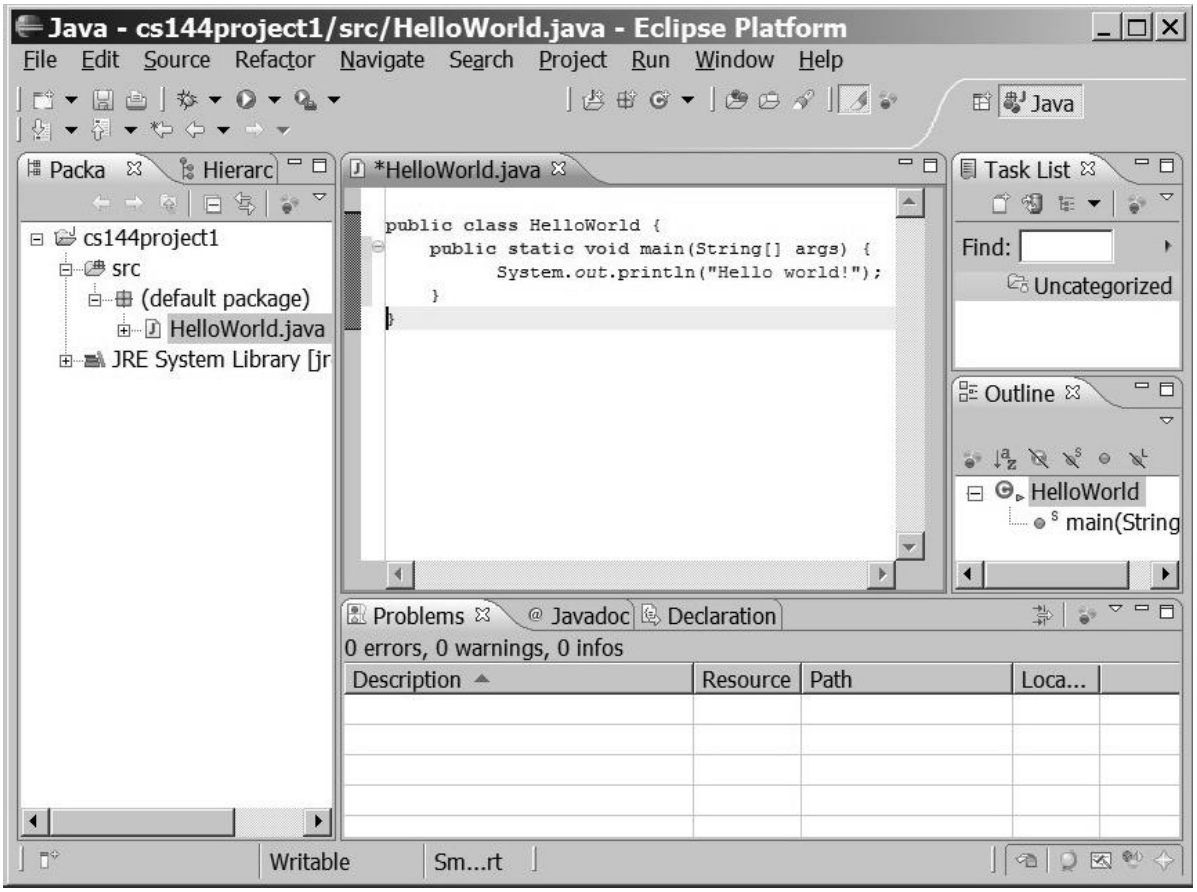


Congratulations! Now you have created your first Java code in eclipse. As you can see from the Package explore pane, your project now includes HelloWorld.java file. The "Editor pane" to the right of the Package explorer pane shows the actual content of the HelloWorld.java file, which simply declares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

Saving, compiling, and running Java code

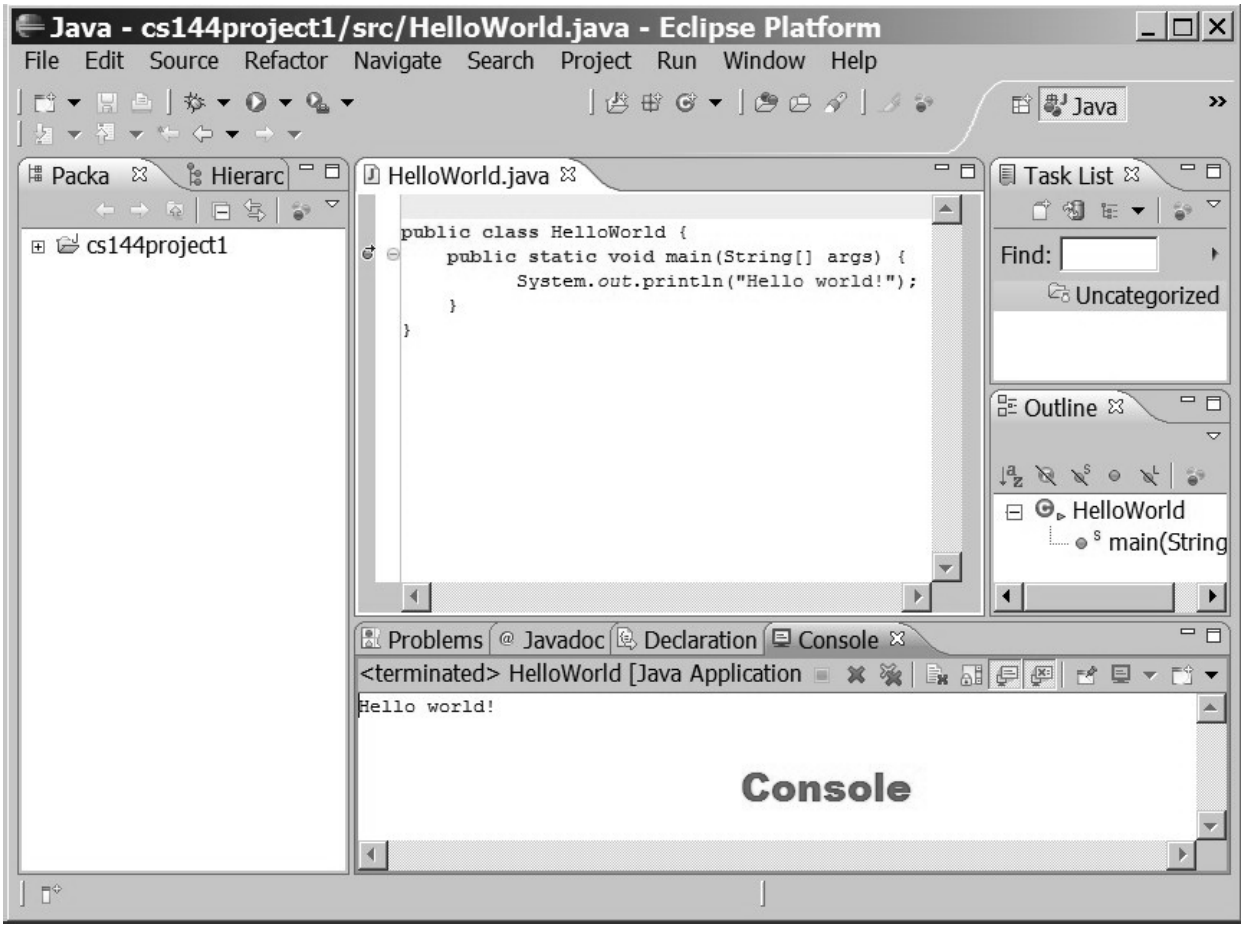
Now let us learn how to code, compile and run a Java program in Eclipse. First copy and paste the following method into the HelloWorld class definition:

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



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.



Quitting Eclipse

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

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

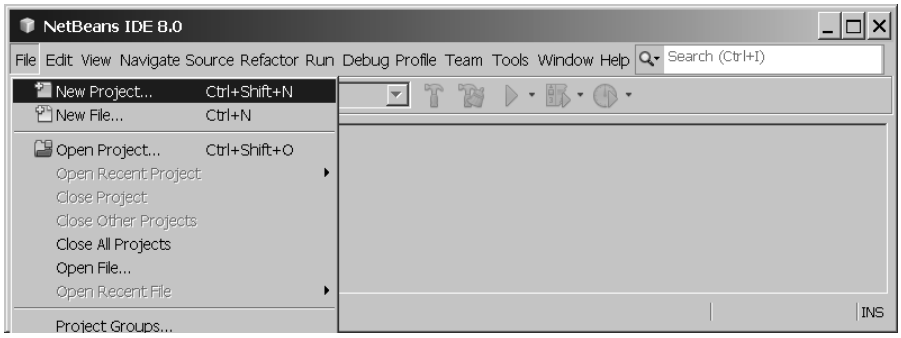
Now you have learned the very basic functionality of Eclipse. Eclipse supports many more functionalities than what you just learned, including integrated debugging and automatic code completion and method lookup, etc. Please read online Eclipse manual to learn more about Eclipse.

Netbeans

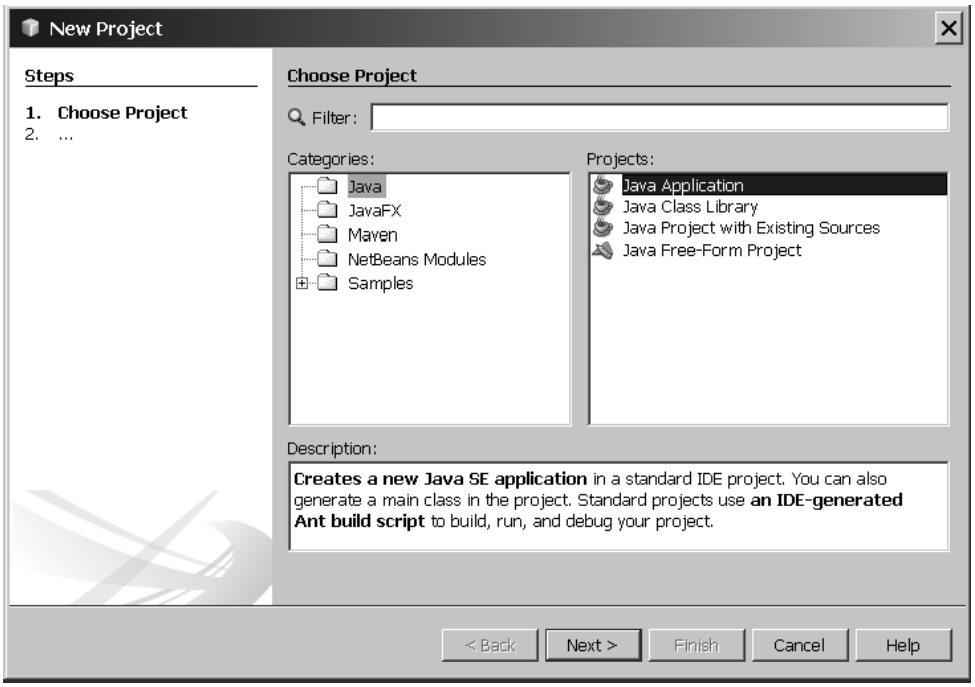
Create an IDE Project

To create an IDE project:

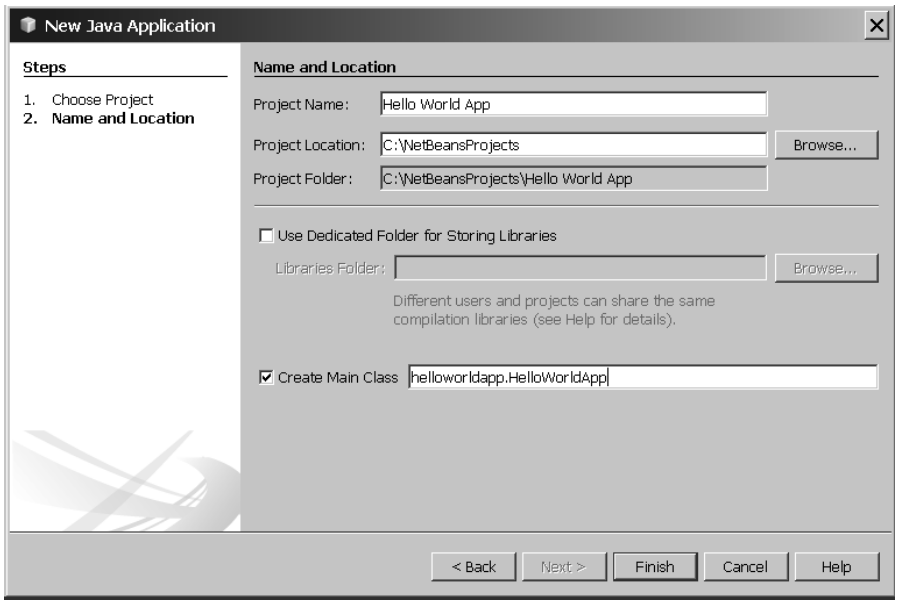
1. Launch the NetBeans IDE.
    - On Microsoft Windows systems, you can use the NetBeans IDE item in the Start menu.
    - On Solaris OS and Linux systems, you execute the IDE launcher script by navigating to the IDE's bin directory and typing ./netbeans.
    - On Mac OS X systems, click the NetBeans 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 **Java Application** as shown in the following figure:



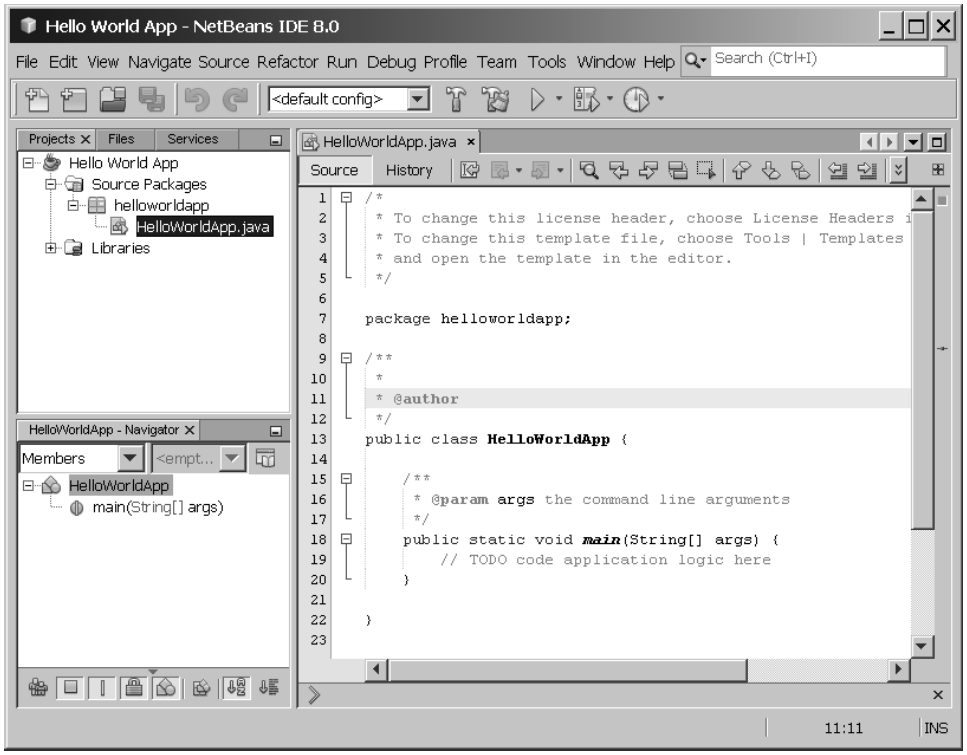
1. NetBeans IDE, New Project wizard, Choose Project page.
2. In the **Name and Location** page of the wizard, do the following (as shown in the figure below):
    - In the **Project Name** field, type Hello World App.
    - In the **Create Main Class** field, type helloworldapp.HelloWorldApp.



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:

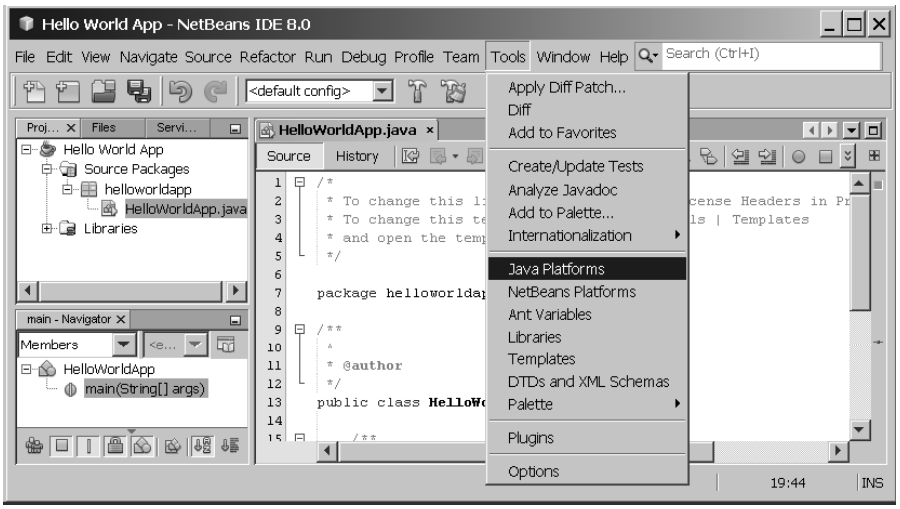
- The **Projects** window, which contains a tree view of the components of the project, including source files, libraries that your code depends on, and so on.
- The **Source Editor** window with a file called HelloWorldApp.java open.
- The **Navigator** window, which you can use to quickly navigate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



- NetBeans IDE with the HelloWorldApp project open.

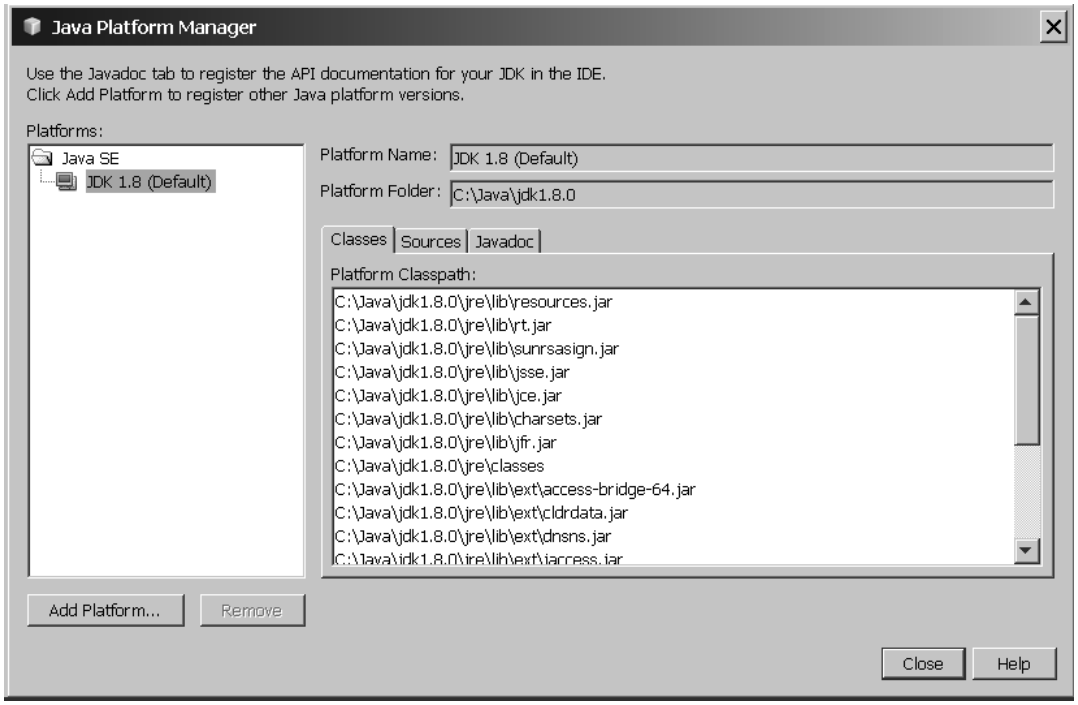
Add JDK 8 to the Platform List (if necessary)

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

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:

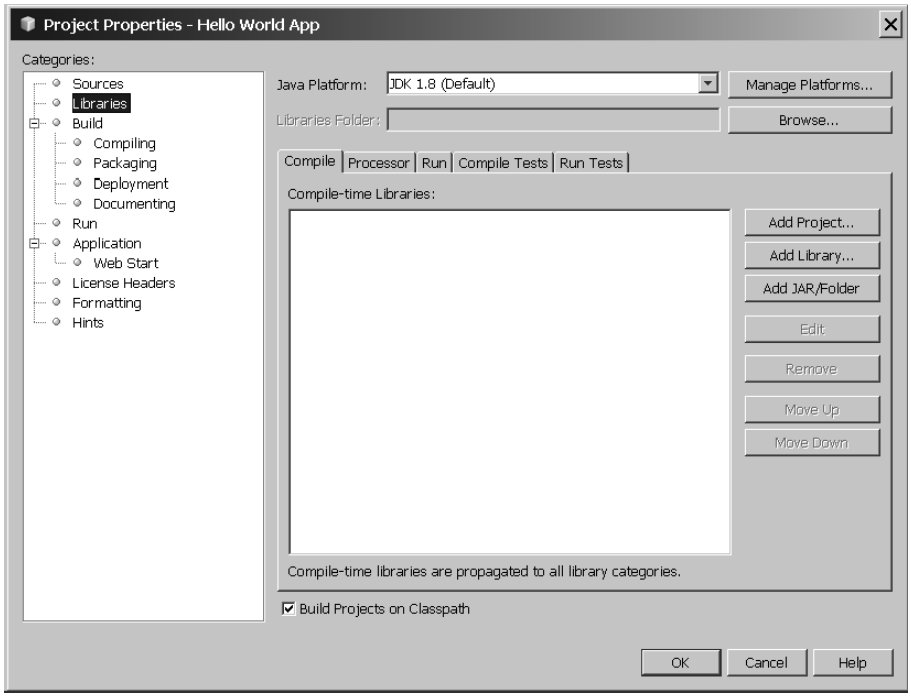


The Java Platform Manager

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 property of your 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 following:



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 here with

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 applicationthat
 * 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:

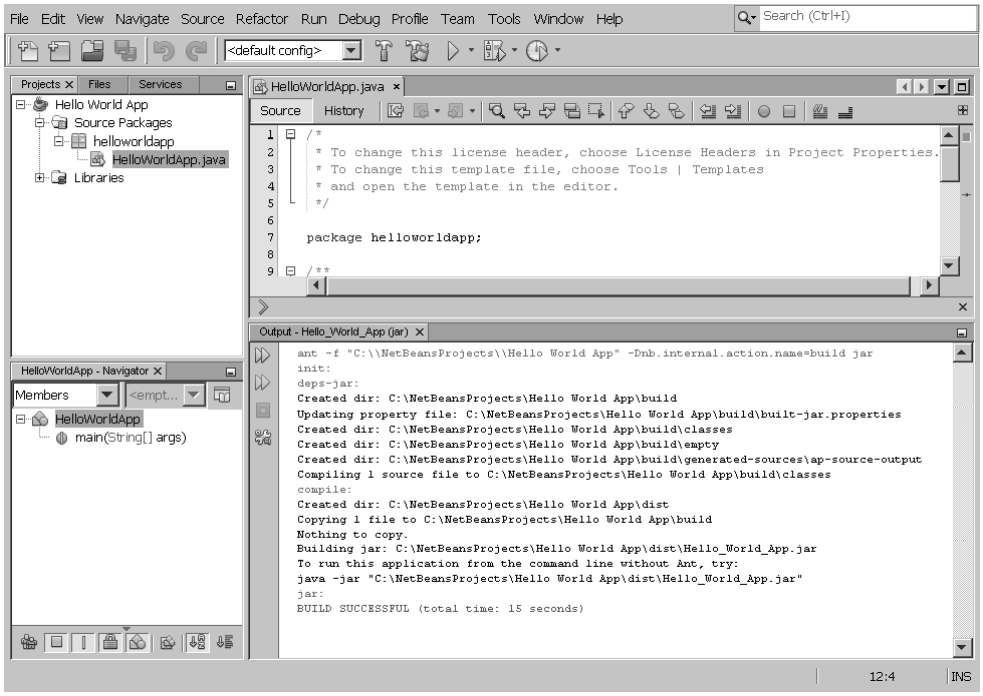
```
/*
 * To change this template, choose Tools |Templates
 * and open the template in the editor.
 */

package helloworldapp;

/**
 * The HelloWorldApp class implements an applicationthat
 * 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:



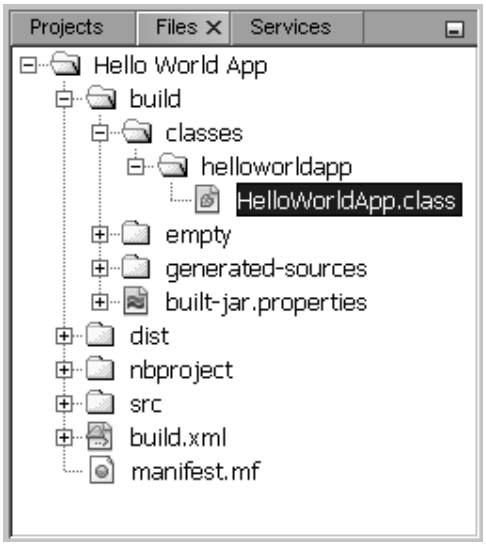
Output window showing results of building the HelloWorld project.

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 expanding

the **Hello World App/build/classes/helloworldapp** node as shown in the following figure.



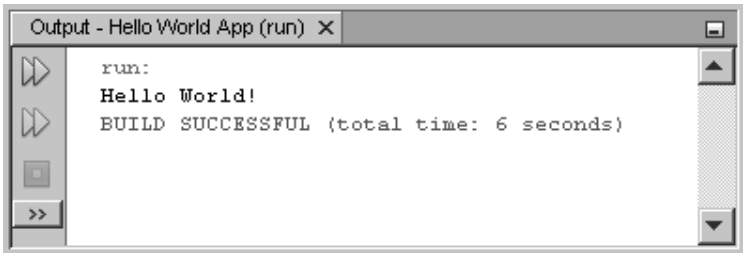
Files window, showing the generated .class file.

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.



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:

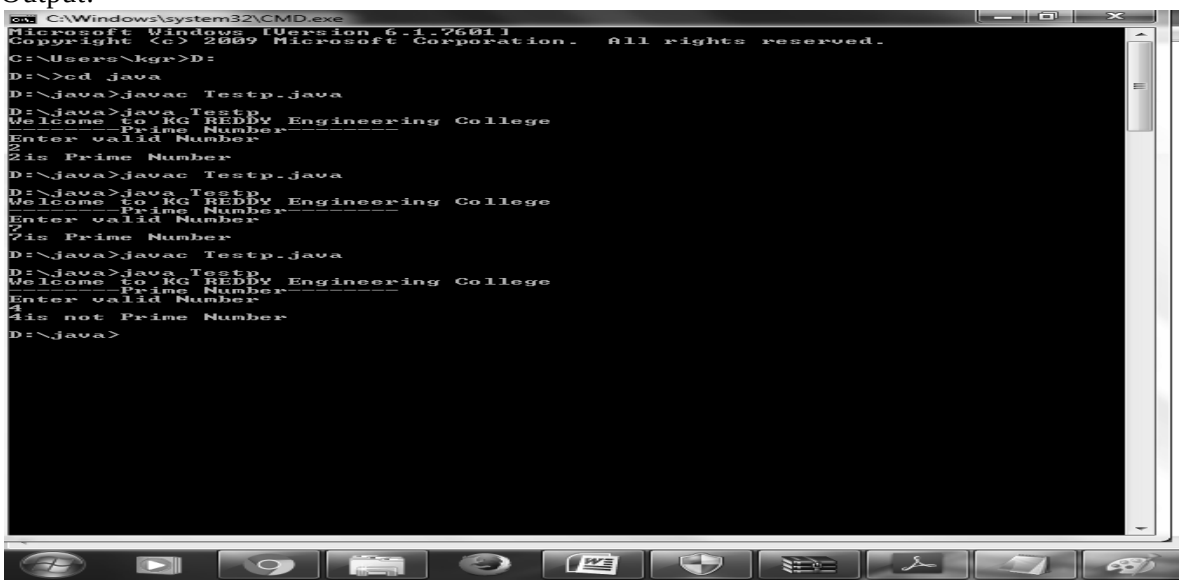
- Once you have created a project in the IDE, you can add files to the project using the **New File** wizard. Choose **File** | **New File**, and then select a template in the wizard, such as the Empty Java File template.
- You can compile and run an individual file (as opposed to a whole project) using 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 Main Project** 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.

1) Use Eclipse or Netbean 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.

Program:

```
import java.util.*;
public class Testp {
public static void main(String[] args) {
System.out.println("Welcome to KG REDDY Engineering College");
System.out.println("--------Prime Number ------ ");
Scanner sc = new Scanner(System.in);
System.out.println("Enter valid Number");
int n = sc.nextInt();
int c = 0;
for (int i = 1; i <= n; i++) {
if (n % i == 0) {
c++;
}
}
if (c == 2) {
System.out.println(n + "is Prime Number");
} else {
System.out.println(n + "is not Prime Number");
}
}
}
```

Output:



2) 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.

AIM : java program that works as a simple calculator.use a Grid layout to arrange buttons for the digits and for the + - * %operations. Add a text field to display the result.

THEORY: GridLayout is one of the Layout managers.A layout manager automatically arranges your controls with in a window by using some type of algorithm.Grid Layout lays out component in a two dimensional grid. When you instantiate a GridLayout,you define the number of rows and columns

Program:

```
import javax.swing.*;
import javax.swing.event.*;
import java.awt.*;
import java.awt.event.*;
class A extends JFrame implements ActionListener {
public JButton b1, b2, b3, b4, b5, b6, b7, b8, b9, b10, b11, b12, b13, b14, b15, b16;
public JTextField tf1;
public JPanel p;
public String v = "";
public String v1 = "0";
public String op = "";
public A() {
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(400, 400);
p = new JPanel(new FlowLayout());
tf1 = new JTextField(10);
p.add(tf1);
add(p);
setLayout(new GridLayout(0,3));
b1 = new JButton("1");
b1.addActionListener(this);
add(b1);
b2 = new JButton("2");
b2.addActionListener(this);
add(b2);
b3 = new JButton("3");
b3.addActionListener(this);
add(b3);
b4 = new JButton("4");
b4.addActionListener(this);
add(b4);
b5 = new JButton("5");
b5.addActionListener(this);
add(b5);
b6 = new JButton("6");
b6.addActionListener(this);
add(b6);
```

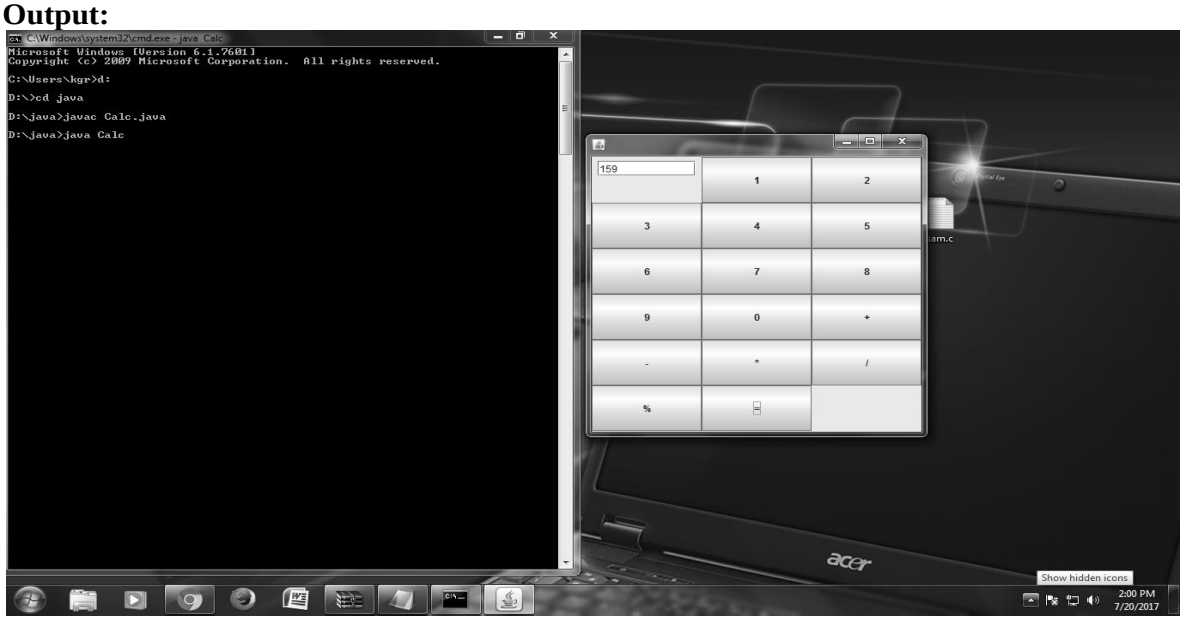
```
b7 = new JButton("7");
b7.addActionListener(this);
add(b7);
b8 = new JButton("8");
b8.addActionListener(this);
add(b8);
b9 = new JButton("9");
b9.addActionListener(this);
add(b9);
b10 = new JButton("0");
b10.addActionListener(this);
add(b10);
b11 = new JButton("+");
b11.addActionListener(this);
add(b11);
b12 = new JButton("-");
b12.addActionListener(this);
add(b12);
b13 = new JButton("*");
b13.addActionListener(this);
add(b13);
b14 = new JButton("/");
b14.addActionListener(this);
add(b14);
b16 = new JButton("%");
b16.addActionListener(this);
add(b16);
b15 = new JButton("=");
b15.addActionListener(this);
add(b15);
setVisible(true);
}
public void actionPerformed(ActionEvent ae) {
String b = ae.getActionCommand();
switch (b) {
case "1": {
v = v + "1";
tf1.setText(v);
}
break;
case "2": {
v = v + "2";
tf1.setText(v);
}
break;
case "3": {
v = v + "3";
tf1.setText(v);
}
break;
```

```
case "4": {
v = v + "4";
tf1.setText(v);
}
break;
case "5": {
v = v + "5";
tf1.setText(v);
}
break;
case "6": {
v = v + "6";
tf1.setText(v);
}
break;
case "7": {
v = v + "7";
tf1.setText(v);
}
break;
case "8": {
v = v + "8";
tf1.setText(v);
}
break;
case "9": {
v = v + "9";
tf1.setText(v);
}
break;
case "0": {
v = v + "0";
tf1.setText(v);
}
break;
case "+": {
op = "+";
v1 = tf1.getText();
v = "";
}
break;
case "-": {
op = "-";
v1 = tf1.getText();
v = "";
}
break;
case "*": {
op = "*";
v1 = tf1.getText();
```

```
v = "";
}
break;
case "/": {
op = "/";
v1 =tf1.getText();
v = "";
}
break;
case "%": {
op = "%";
v1 = tf1.getText();
v = "";
}
break;
case "=": {
switch (op) {
case "+": {
v = tf1.getText();
if (v.equals("")) {
v = "0";
}
long i = Long.parseLong(v1) + Long.parseLong(v);
tf1.setText(String.valueOf(i));
v="";
}
break;
case "-": {
v = tf1.getText();
if (v.equals("")) {
v = "0";
}
long i = Long.parseLong(v1) - Long.parseLong(v);
tf1.setText(String.valueOf(i));
v="";
}
break;
case "*": {
v = tf1.getText();
if (v.equals("")) {
v = "0";
}
long i = Long.parseLong(v1) * Long.parseLong(v);
tf1.setText(String.valueOf(i));
v="";
}
break;
case "/": {
try {
v = tf1.getText();
```

```
if (v.equals("")) {
v = "0";
}
long i = Long.parseLong(v1) / Long.parseLong(v);
tf1.setText(String.valueOf(i));
v="";
} catch (Exception ex) {
JOptionPane.showMessageDialog(this, ex.getMessage());
}
}
break;
case "%": {
try {
v = tf1.getText();
if (v.equals("")) {
v = "0";
}
long i = Long.parseLong(v1) % Long.parseLong(v);
tf1.setText(String.valueOf(i));
v="";
} catch (Exception ex) {
JOptionPane.showMessageDialog(this, ex.getMessage());
}
}
break;
}
}
break;
}
}
}
public class Calc {
public static void main(String[] args) {
A a = new A();
}
}
```

Output:


Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\kgr>d:
D:\>cd java
D:\java>javac Calc.java
D:\java>java Calc
159
1
2
3
4
5
6
7
8
9
0
+
-
*
/
%
=
acer
Show hidden icons
2:00 PM
7/20/2017

3.a) Develop an Applet in java that displays a Simple Message.

Aim : Java program to display a simple message.

Theory: Applets are designed to bring the web alive.they function to add animation sound and eventually complete multi media into HTML documents.java is also part of the future of interfacing with virtual-reality environments implemented via VRML.At present ,java is limited only by the capabilities of the internet itself.applets are java programs that are specialized for use over the Web.

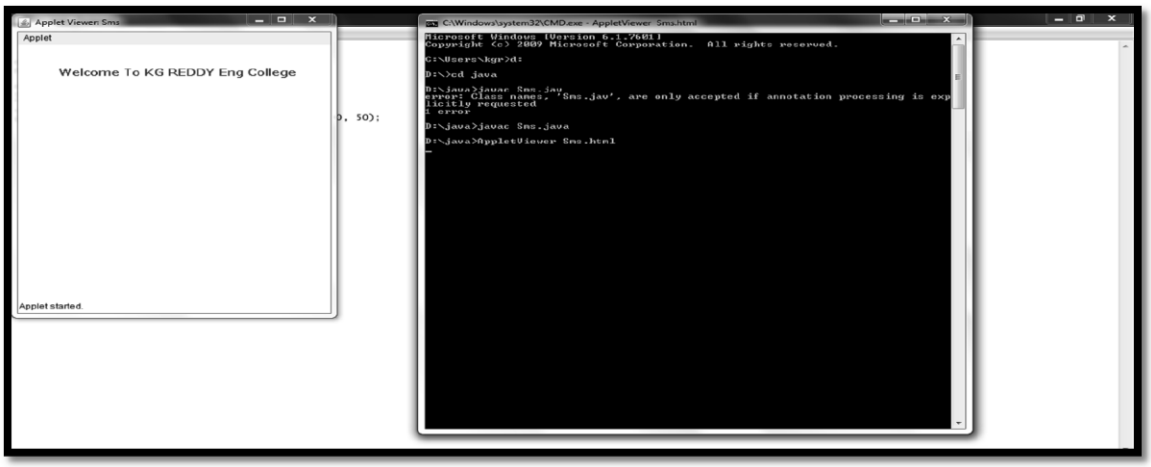
Program:

```
import java.applet.Applet;
import java.awt.*;
public class Sms extends Applet {
public void init() {
}
public void paint(Graphics g) {
g.setColor(Color.blue);
Font font = new Font("verdana", Font.BOLD, 15);
g.setFont(font);
g.drawString("Welcome To KG REDDY Eng College", 50, 50);
}
}
```

Applet code:

```
<html>

<body>

<applet code="Sms" width=45 height=54>

</applet>

</body>

</html>
```

Output:



3.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.

Aim : Develop an applet which receives an integer in one text field, and computes its factorial Value and returns it in another text field.

Theory: Applets are designed to bring the web alive.they function to add animation sound and eventually complete multi media into HTML documents.java is also part of the future of interfacing with virtual-reality environments implemented via VRML.At present ,java is limited only by the capabilities of the internet itself.applets are java programs that are specialized for use over the Web. The Applet life cycle

The init()Method: The init()method is where your applet does much of its setup,such as defined its layout,parsing parameters,or setting the background colors.
The starts() Method: The start()method is used mainly when implementing threads in java.
The stop() Mehtod: The stop() method is used to do what its name suggests: stop what is going on.
The destroy() method: when it is called,the applet is told to free up system resources.

Program:

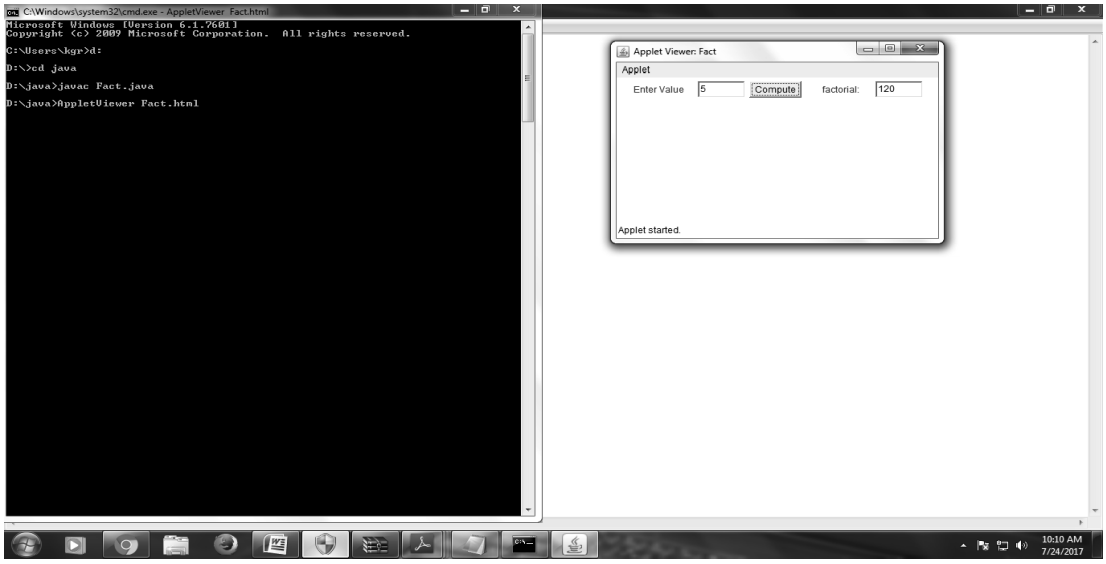
```
import java.applet.Applet;

import java.awt.*;

import java.awt.event.*;

public class Fact extends Applet implements ActionListener {

Label l1, l2, l3;

TextField tf1, tf2;

Button b1;

public void init() {

setSize(400, 200);

FlowLayout g = new FlowLayout();

setLayout(g);

l1 = new Label("Enter Value");

l1.setForeground(Color.BLUE);

add(l1);

tf1 = new TextField(5);

tf1.setText("0");
```

```
add(tf1);
b1 = new Button("Compute");
b1.addActionListener(this);
add(b1);
l3 = new Label();
add(l3);
l2 = new Label("factorial: ");
l2.setForeground(Color.BLUE);
add(l2);
tf2 = new TextField(5);
add(tf2);
}
public void actionPerformed(ActionEvent ae) {
long n = Integer.parseInt(tf1.getText());
long f = 1;
while (n != 0) {
f = f * n;
n--;
}
tf2.setText(String.valueOf(f));
}
}
```

Applet code:

```
 <html>
<body>
<applet code="Fact" width=45 height=54>
</applet>
</body>
</html>
```

Output:


C:\Windows\system32\cmd.exe - AppletViewer Fact.html
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\kgr>d:
D:\>cd java
D:\java>javac Fact.java
D:\java>AppletViewer Fact.html
Applet Viewer: Fact
Applet
Enter Value
5
Compute
factorial:
120
Applet started.
10:10 AM
7/24/2017

4) 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 NumberFormatException, If Num2 is Zero, and the program would throw an Athematic Exception. Display the Exception in message box.

Aim : Program to create a user interface to perform integer divisions. The user enters two numbers in the textfields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an ArithmeticException. Display the exception in a message dialog box.

Theory: The AWT supports a rich assortment of graphics methods.All graphics are drawn relative to a window.this can the main windowof an applet, a child window of an applet,or a stand alone application window. The origin of each window is at the top-left corner and is 0,0 cordinates are specified in pixels.All output to a window takes place through a graphics context.

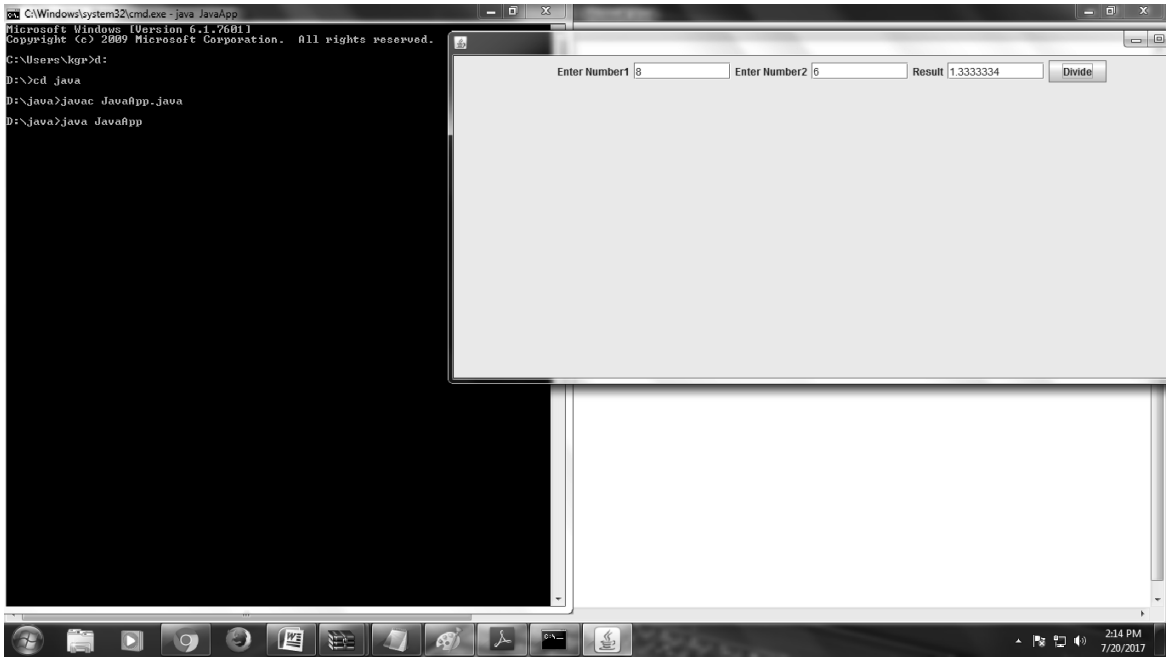
Program:

```
import java.awt.*;

import java.awt.event.*;

import javax.swing.*;

import javax.swing.event.*;

class A extends JFrame implements ActionListener {

JLabel l1, l2, l3;

JTextField tf1, tf2, tf3;

JButton b1;

A() {

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

setLayout(new FlowLayout());

l1 = new JLabel("Welcome");

setSize(800, 400);

l1 = new JLabel("Enter Number1");

add(l1);

tf1 = new JTextField(10);

add(tf1);
```

```
l2 = new JLabel("Enter Number2");
add(l2);
tf2 = new JTextField(10);
add(tf2);
l3 = new JLabel("Result");
add(l3);
tf3 = new JTextField(10);
add(tf3);
b1 = new JButton("Divide");
add(b1);
b1.addActionListener(this);
setVisible(true);
}
public void actionPerformed(ActionEvent ae) {
try {
int a = Integer.parseInt(tf1.getText());
int b = Integer.parseInt(tf2.getText());
if(b==0)
throw new ArithmeticException(" Divide by Zero Error");
float c = (float) a / b;
tf3.setText(String.valueOf(c));
} catch (NumberFormatException ex) {
JOptionPane.showMessageDialog(this,   ex.getMessage());
} catch (ArithmeticException ex) {
JOptionPane.showMessageDialog(this,   ex.getMessage());
}
}
}
public class JavaApp {
```

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

Output:



5) 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 ofnumber.

Aim : Creating a Java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number

Theory: The java run-time system depends on the threads for many things, and all the class libraries are designed with multithreading in mind. In fact,java uses threads to enable the entire environment to be asynchronous.This helps reduce inefficiency by preventing the waste of CPU cycles. The benefits of java's multithreading is that the main loop/polling mechanism is eliminated.one thread can pause without stopping other parts of your program.when a thread blocks in ajava program,only the single thread that is blocked pauses.All other threads continue to run.

Program:

```
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 < 50; 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[] args) {
A a = new A("One");
a.start();
}
}
```

Output:

6. Write a Java program for the following:

i. **Create a doubly linked list of elements.**
ii. **Delete a given element from the above list.**
iii. **Display the contents of the list after deletion.**

Aim : create a program in which doubly linked list operations can be shown in detail.

Theory : A doubly linked list is a linked data structure that consists of a set of sequentially linked records called nodes. Each node contains two fields, called links, that are references to the previous and to the next node in the sequence of nodes.

Program:

```
import java.io.*;

class node

{

public int x;

public node next;

public node prev;

}

class DoubleLinkedList

{

public node first;

public node last;

DoubleLinkedList()

{

first=new node();

first.next=null;

first.prev=null;

last=first;

}
```

```
void add (int v)
{
node temp=new node();
temp.x=v;
temp.next=null;
last.next=temp;
temp.prev=last;
last=temp;
}
void insert(int p,int v)
{
node ptr=first,temp;
for(int i=1;i<=p-1;i++)
ptr=ptr.next;
if(ptr.next==null)
add(v);
else
{
temp=new node();
temp.x=v;
temp.next=ptr.next;
ptr.next.prev=temp;
ptr.next=temp;
temp.prev=ptr;
}
```

```
}
void del(int p)
{
node ptr=first,temp;
for(int i=1;i<=p-1;i++)
ptr=ptr.next;
if(ptr.next.next==null)
{
temp=last;
last=last.prev;
last.next=null;
}
else
{
temp=ptr.next;
ptr.next=ptr.next.next;
ptr.next.prev=ptr;
}
temp=null;
}
void show()
{
System.out.println("\nList Elements:Left to Right");
for(node ptr=first.next;ptr!=null;ptr=ptr.next)
System.out.print("\t"+ptr.x);
```

```
System.out.println("\nList Elements:Right to Left");
for(node ptr=last;ptr.prev!=null;ptr=ptr.prev)
System.out.print("\t"+ptr.x);
}
}
Class DListTest
{
public static void main(String args[]) throws Exception
{
String con="";
int x,op,p,v;
DoubleLinkedList l1=new DoubleLinkedList();
InputStreamReader isr=new InputStreamReader(System.in);
BufferedReader br=new BufferedReader(isr);
System.out.println("Enter elements to create");
do
{
x=Integer.parseInt(br.readLine());
l1.add(x);
System.out.print("Add more?(y,n):");
con=br.readLine();
}while(con.equals("y"))
l1.show();
do
{
```

```
System.out.println("\n 1.Insert\n 2.Delete \n 3.Display \n 4.Exit");
System.out.println("\nSelect an option:");
op=Integer.parseInt(br.readLine());
if(op==1)
{
System.out.println("Enter Position to insert:");
p= Integer.parseInt(br.readLine());
System.out.println("Enter Value to insert:");
v= Integer.parseInt(br.readLine());
l1.insert(p,v);
}
if(op==2)
{
System.out.println("Enter Position to delete:");
p= Integer.parseInt(br.readLine());
l1.del(p);
}
l1.show();
}while(op<4)
}
}
```

Output:

```
Original Linked list
10 8 4 2
Modified Linked List
8
```

7) Write a Java Program that simulates a Traffic Light. The program lets the use select one of three lights :red, yellow or Green with radiobuttons.On selecting radio button, an appropriate message with "stop" or "Ready" or "GO" should appear above the button in selected color.Intially ,there is no message shown.

Aim: write a java program that simulates trafficlight the program let user select one of three lights ,thread yellowor green .when a radio button is select the light is turn ed one light can be on at atime. No lights is on when the program starts.

Theory: The AWT supports a rich assortment of graphics methods.All graphics are drawn relative to a window.this can the main windowof an applet, a child window of an applet,or a stand alone application window. The origin of each window is at the top-left corner and is 0,0 cordinates are specified in pixels.All output to a window takes place through a graphics context.

Program:

```
import javax.swing.*;
import javax.swing.event.*;
import java.awt.*;
import java.awt.event.*;
class A extends JFrame implements ItemListener {
public JLabel l1, l2;
public JRadioButton r1, r2, r3;
public ButtonGroup bg;
public JPanel p, p1;
public A() {
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLayout(new GridLayout(2, 1));
setSize(800,  400);
p = new JPanel(new FlowLayout());
p1 = new JPanel(new FlowLayout());
l1 = new JLabel();
Font f = new Font("Verdana", Font.BOLD, 60);
l1.setFont(f);
add(l1);
p.add(l1);
add(p);
l2 = new JLabel("Select Lights");
p1.add(l2);
JRadioButton r1 = new JRadioButton("Red Light");
r1.setBackground(Color.red);
p1.add(r1);
r1.addItemListener(this);
JRadioButton r2 = new JRadioButton("Yellow Light");
r2.setBackground(Color.YELLOW);
p1.add(r2);
r2.addItemListener(this);
JRadioButton r3 = new JRadioButton("Green Light");
r3.setBackground(Color.GREEN);
p1.add(r3);
```

```
r3.addItemListener(this);
add(p1);
bg = new ButtonGroup();
bg.add(r1);
bg.add(r2);
bg.add(r3);
setVisible(true);
}
public void itemStateChanged(ItemEvent i) {
JRadioButton jb = (JRadioButton) i.getSource();
switch (jb.getText()) {
case "Red Light": {
l1.setText("STOP");
l1.setForeground(Color.red);
}
break;
case "Yellow Light": {
l1.setText("Ready");
l1.setForeground(Color.YELLOW);
}
break;
case "Green Light": {
l1.setText("GO");
l1.setForeground(Color.GREEN);
}
break;
}
}
}
public class TLights {
public static void main(String[] args) {
A a = new A();
}
}
```

Output:

8) 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 subclass 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 Shape.

Aim: Write a java program to create an abstract class that illustrates different geometrical figures.

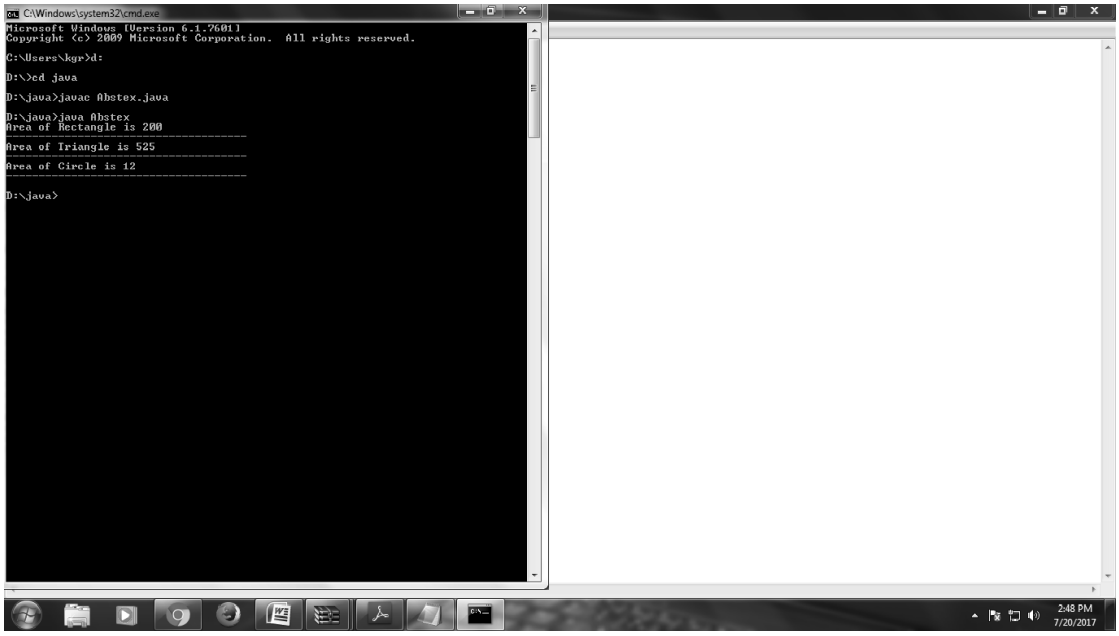
Theory: To create an abstract class that shows the hiding of elements in a class. At the same time inheritance property is used to extend the class shape into different geometrical figures. This represents the reusability of code for a programmer.

Program:

```
abstract class shape {
public int x, y;
public abstract void printArea();
}
class Rectangle extends shape {
public void printArea() {
System.out.println("Area of Rectangle is " + x * y);
}
}
class Triangle extends shape {
public void printArea() {
System.out.println("Area of Triangle is " + (x * y) / 2);
}
}
class Circle extends shape {
public void printArea() {
System.out.println("Area of Circle is " + (22 * x * x) / 7);
}
}
public class Abstex {
/**
* @param args the command line arguments
*/
public static void main(String[] args) {
// TODO code application logic here
Rectangle r = new Rectangle();
r.x = 10;
r.y = 20;
r.printArea();
System.out.println(" ------------------------------------");
Triangle t = new Triangle();
t.x = 30;
t.y = 35;
t.printArea();
```

```
System.out.println(" ------------------------------------");
Circle c = new Circle();
c.x = 2;
c.printArea();
System.out.println(" ------------------------------------");
}
}
```

Output:



9) 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 rows in table. The elements are separated by commas. Write java program to display the table using Label in Grid Layout.

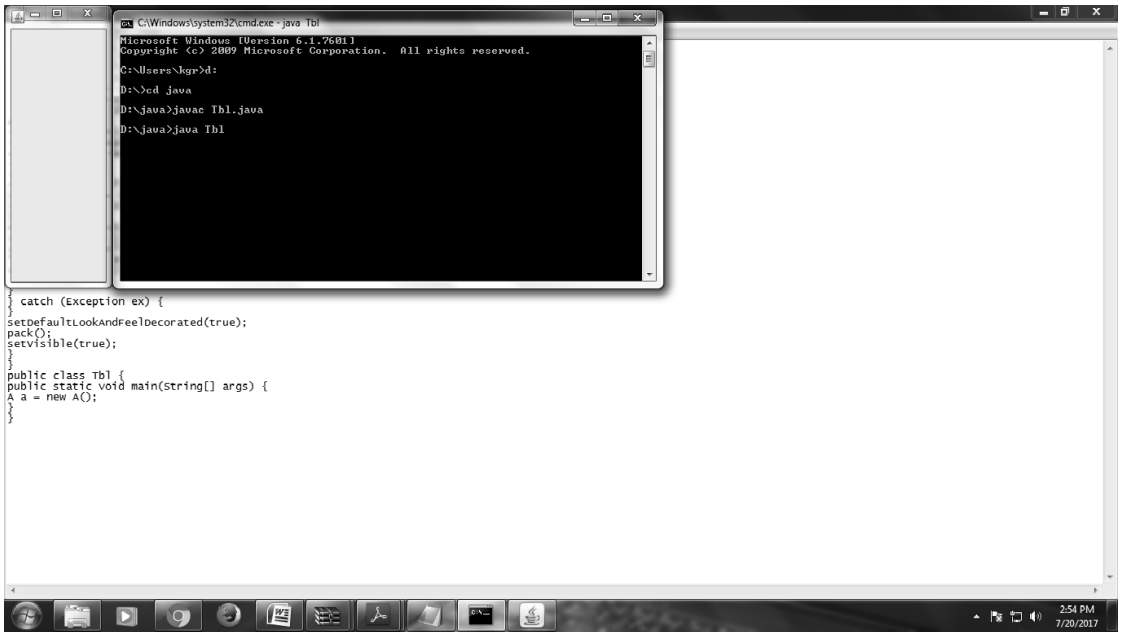
Aim: Write a java program to display the table using Grid Layout component

Theory: To create an table and display it using JTable components

Program:

```
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("D:\\emp.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) {
}
setDefaultLookAndFeelDecorated(true);
pack();
setVisible(true);
}
}
public class Tbl {
public static void main(String[] args) {
A a = new A();
}
```

Output:


C:\Windows\system32\cmd.exe - java Tbl
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\kgr>d:
D:\>cd java
D:\java>javac Tbl.java
D:\java>java Tbl
} catch (Exception ex) {
}
setDefaultLookAndFeelDecorated(true);
pack();
setVisible(true);
}
}
public class Tbl {
public static void main(String[] args) {
A a = new A();
}
}
2:54 PM
7/20/2017

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

- **Aim :** Write a java program for handling mouse events

Theory : To handle mouse events you must implement the MouseListener and the MouseMotionListener interfaces.These two interfaces contain methods that receive and process the various types of mouse events.

Program:

```
import javax.swing.*;
import java.awt.*;
import javax.swing.event.*;
import java.awt.event.*;
class A extends JFrame implements MouseListener {
JLabel l1;
public A() {
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(400, 400);
setLayout(new GridBagLayout());
l1 = new JLabel();
Font f = new Font("Verdana", Font.BOLD, 20);
l1.setFont(f);
l1.setForeground(Color.BLUE);
l1.setAlignmentX(Component.CENTER_ALIGNMENT);
l1.setAlignmentY(Component.CENTER_ALIGNMENT);
add(l1);
addMouseListener(this);
setVisible(true);
}
public void mouseExited(MouseEvent m) {
l1.setText("Mouse Exited");
}
public void mouseEntered(MouseEvent m) {
l1.setText("Mouse Entered");
}
public void mouseReleased(MouseEvent m) {
l1.setText("Mouse Released");
}
public void mousePressed(MouseEvent m) {
l1.setText("Mouse Pressed");
}
public void mouseClicked(MouseEvent m) {
l1.setText("Mouse Clicked");
}
}
public class Mevents {
public static void main(String[] args) {
A a = new A();
```

```
}
}
```

Output:


C:\Windows\system32\cmd.exe - java Mevents
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\kgr>d:
D:\>cd java
D:\java>javac Mevents.java
D:\java>java Mevents
Mouse Exited
2:59 PM
7/20/2017

11) 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)

Aim: Write a java program which can read text file into hash table and print the hash values based on the hash key input.

THEORY: Text file will contain names and phone numbers which are separated by a tab.

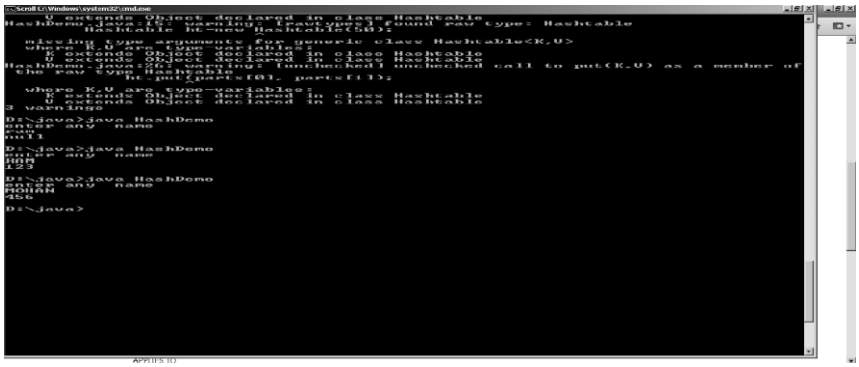
This information has to be recorded in to hash table.

Program:

```
import java.util.*;
import java.io.*;
public class Hashtbl {
public static void main(String[] args) {
try {
FileInputStream fs = new FileInputStream("D:\\ph.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 KG REDDY Eng College");
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 KG REDDY Eng College");
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 betwen 1 and Three");
break;
}
}
} catch (Exception ex) {
System.out.println(ex.getMessage());
}
}
}
```

Output:



12) Write a Java program that correctly implements the producer – consumer problem using the concept of interthread communication.

Aim: program that implements producer-consumer problem.

Theory : In computing, the producer–consumer problem (also known as the bounded-buffer problem) is a classic example of a multi-process synchronization problem. The problem describes two processes, the producer and the consumer, which share a common, fixed-size buffer used as a queue.

- The producer's job is to generate data, put it into the buffer, and start again.
- At the same time, the consumer is consuming the data (i.e. removing it from the buffer), one piece at a time

Program:

```
class Q
{
int n;
boolean valueSet=false;
synchronized int get()
{
if(!valueSet)
try
{
wait();
}
catch(InterruptedException e)
{

System.out.println("Interrupted Exception caught");
}
System.out.println("Got:"+n);
valueSet=false;
notify();
return n;
}
synchronized void put(int n)
{
if(valueSet)
try
{
wait();
}
catch(InterruptedException e)
{
System.out.println("Interrupted Exception caught");
}
this.n=n;
valueSet=true;
```

```
System.out.println("Put:"+n);
notify();
}
}
class Producer implements Runnable
{
Q q;
Producer(Q q)
{
this.q=q;
new Thread(this,"Producer").start();
}
public void run()
{
int i=0;
while(true)
{
q.put(i++);
}
}
}
class Consumer implements Runnable
{
Q q;
Consumer(Q q)
{
this.q=q;
new Thread(this,"Consumer").start();
}
public void run()
{
while(true)
{
q.get();
}
}
}

class ProdCons
{
public static void main(String[] args)
{
Q q=new Q();
new Producer(q);
newConsumer(q);
System.out.println("Press Control-c to stop");
}
}
```

Output:


C:\Windows\system32\cmd.exe - java ProdCons
Got:39201
Put:39202
Got:39202
Put:39203
Got:39203
Put:39204
Got:39204
Put:39205
Got:39205
Put:39206
Got:39206
Put:39207
Got:39207
Put:39208
Got:39208
Put:39209
Got:39209
Put:39210
Got:39210
Put:39211
Got:39211
Put:39212
Got:39212
Put:39213
Got:39213
Put:39214
Got:39214
Put:39215
Got:39215
Put:39216
Got:39216
Put:39217
Got:39217
Put:39218
Got:39218
Put:39219
Got:39219
Put:39220
Got:39220
Put:39221
Got:39221
Put:39222
Got:39222
Put:39223
Got:39223
Put:39224
Got:39224
Put:39225
Got:39225
Put:39226
Got:39226
Put:39227
Got:39227
Put:39228
Got:39228
Put:39229
Got:39229
sam.c
acer
acer
3:28 PM
7/20/2017

13) Write a Java program to list all the files in a directory including the files present in all its subdirectories.

Aim: java program to list all the files in a directory.

Theory : Java provides a standard way of reading from and writing to files. Traditionally thejava.io package was used, but in modern Java applications you use the java.nio.fileAPI. Java will read all input as a stream of bytes. The InputStream class is the superclass of all classes representing an input stream of bytes.

Program:

```
import java.io.*;

import java.lang.*;

import java.io.File;

public class ListFilesUtil {

   public void listFilesAndFolders(String directoryName){

      File directory = new File(directoryName);

      //get all the files from a directory

      File[] fList = directory.listFiles();

      for (File file : fList){

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

      }

   }

   public void listFiles(String directoryName){

      File directory = new File(directoryName);

      //get all the files from a directory

      File[] fList = directory.listFiles();

      for (File file : fList){

         if                    (file.isFile()){

            System.out.println(file.getName());
```

```
        }
    }
}
public void listFolders(String directoryName){
    File directory = new File(directoryName);
    //get all the files from a directory
    File[] fList = directory.listFiles();
    for (File file : fList){
        if                    (file.isDirectory()){
            System.out.println(file.getName());
        }
    }
}
public  void  listFilesAndFilesSubDirectories(String  directoryName){
    File directory = new File(directoryName);
    //get all the files from a directory
    File[] fList = directory.listFiles();
    for (File file : fList){
        if (file.isFile()){
            System.out.println(file.getAbsolutePath());
        } else if (file.isDirectory()){
            listFilesAndFilesSubDirectories(file.getAbsolutePath());
        }
    }
}
```

```
    public static void main(String[] args){
        ListFilesUtil listFilesUtil = new ListFilesUtil();
        final String directoryLinuxMac ="/Users/loiane/test";
        final String directoryWindows ="C://test";
        listFilesUtil.listFiles(directoryLinuxMac);
    }
}
```

Output:

```
*********************************************
Files from main directory : C:\Users\Gaurav Miglani\Desktop\Test
*********************************************
Cormen.pdf
Extra-Items.pdf
XYZ.pdf
[Docs]
    A.docx
    B.doc
    C.docx
ABC.pdf
JKL.pdf
[sheets]
    XXX.csv
    YYY.csv
results.pdf
[Resumes]
    [Before2016]
        Resume2015.doc
        Resume2016.doc
        [Before2014]
            Resume2014.doc
    Resume2017.doc
    Resume2017.pdf
        QA.doc
Testing.pdf
```

14) Write a java program that implements QuickSort algorithm for sorting a list of names in ascending order.

Aim: java program to implement quicksort algorithm in ascending order.

Theory : QuickSort is a Divide and Conquer algorithm. It picks an element as pivot and partitions the given array around the picked pivot. There are many different versions of quickSort that pick pivot in different ways.

1. Always pick first element as pivot.
2. Always pick last element as pivot (implemented below)
3. Pick a random element as pivot.
4. Pick median as pivot.

Program :

```
public class MyQuickSort {

private int array[];

private int length;

public void sort(int[] inputArr) {

if (inputArr == null || inputArr.length == 0) {

return;

}

this.array = inputArr;

length = inputArr.length;

quickSort(0, length - 1);

}

private void quickSort(int lowerIndex, int higherIndex) {

int i = lowerIndex;

int j = higherIndex;

int pivot = array[lowerIndex+(higherIndex-lowerIndex)/2];

while (i <= j) {

while (array[i] < pivot) {

i++;
```

```
}
while (array[j] > pivot) {
j--;
}
if (i <= j) {
exchangeNumbers(i, j);
i++;
j--;
}
}
if (lowerIndex < j)
quickSort(lowerIndex, j);
if (i < higherIndex)
quickSort(i, higherIndex);
}
private void exchangeNumbers(int i, int j) {
int temp = array[i];
array[i] = array[j];
array[j] = temp;
}
public static void main(String a[]){
MyQuickSort sorter = new MyQuickSort();
int[] input = {24,2,45,20,56,75,2,56,99,53,12};
sorter.sort(input);
for(int i:input){
```

```
System.out.print(i);

System.out.print(" ");

}

}

}
```

Output:

15) Write a Java program that implements Bubble sort algorithm for sorting in descending order and also shows the number of interchanges occurred for the given set of integers.

Aim: java program to implements bubble sort algorithm

Theory : Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order.

Program:

```
import java.util.Scanner;
class BubbleSort {
public static void main(String []args) {
int num, i, j, temp;
Scanner input = new Scanner(System.in);
System.out.println("Enter the number of integers to sort:");
num = input.nextInt();
int array[] = new int[num];
System.out.println("Enter " + num + " integers: ");
for (i = 0; i < num; i++)
array[i] = input.nextInt();
for (i = 0; i < ( num - 1 ); i++) {
for (j = 0; j < num - i - 1; j++) {
if (array[j] < array[j+1])
{
temp = array[j];
array[j] = array[j+1];
array[j+1] = temp;
}
}
}
System.out.println("Sorted list of integers:");
for (i = 0; i < num; i++)
System.out.println(array[i]);
}
}
```

Output: