1.OBJECT ORIENTED PROGRAMMING FUNDAMENTALS

Example #1: A simple java Program

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

The above program prints a message “hello world” on stand output.

Steps to write and execute Java Program

1. Open an editor like Notepad, MS-DOS editor or editplus
2. Type source code
3. Compile code
4. Execute it

Setting PATH, CLASSPATH

Desktop → Right click on ‘Computer’→ Properties → Advanced Change Settings → Advanced → Environment Variables → User Variables (OR) System Variables

PATH = .;C:\Program Files\Java\jdk1.6.0_18\bin

CLASSPATH=.;C:\Program Files\Java\jdk1.6.0_18\jre\lib\rt.jar

Compilation & Execution

1. Open DOS-Prompt
2. Compiling program
   Syntax:
   Java FileName .java
3. Executing program
   Syntax:
   Java FileName

Example #2: Talking Input from command line

```
public class sum {
        public static void main(String[] args) {
                int a, b, sum;
                a=Integer.parseInt(args[0]);
               b= Integer.parseInt(args[1]);
                sum=a+b;
                System.out.println("The sum is" +sum);

        }
}
```

The above program takes two integers through command line and prints the sum of them.

Run the program as follows:
Java Sum 10 20
O/P:
The sum is 30

Example #3: Printing multiplication for the given number

```
class Table {
        public static void main(String args[]){
                int i = 1, n;
                n = Integer.parseInt(args[0]);
                while (i <= 10) {
                        System.out.println(n + "*" + i + "=" + (n * i));
                        i++;
                }
        }
}
C:\aspire>java Table 5
```

Example #4: Finding whether the given numbers is a palindrome:

```
class Palindrome {
        public static void main(String args[]) {
                int no, no1, digit, rev = 0;
                no = Integer.parseInt(args[0]);
                no1 = no;
                while (no > 0) {
                        digit = no % 10;
                        rev = rev * 10 + digit;
                        no /= 10;
                }
                if (rev == no1)
                        System.out.println("palindrome");
                else
                        System.out.println("Not palindrome");
        }

}
```

Example #5: Program to spell out the given number

```
import java.io.*;
class Spell {
        Public static void main(String[] args){
                int no = Integer.parseInt(args[0]);
```

```
        int digit, reverse = 0;
        for (; no>0; no /= 10) {
                digit = no % 10;
                reverse = reverse * 10 + digit;
        }
        for (; reverse > 0; reverse /= 10) {
                digit = reverse % 10;
                switch (digit) {
                case 0:
                        System.out.print("Zero");
                        break;
                case 1:
                        System.out.print("One");
                        break;
                case 2:
                        System.out.print("Two");
                        break;
                case 3:
                        System.out.print("Three");
                        break;
                case 4:
                        System.out.print("Four");
                        break;
                case 5:
                        System.out.print("Five");
                        break;
                case 6:
                        System.out.print("Six");
                        break;
                case 7:
                        System.out.print("Seven");
                        break;
                case 8:
                        System.out.print("Eight");
                        break;
                case 9:
                        System.out.print("Nine");
                        break;
                }
        }
    }
}
```

Enhanced for loop(for each loop)

It is a new feature added by J2SE 5. This is most convenient loop for arrays and collections.

Syntax:

```
for(<declaration> : <collectios/array>)  {
    //code
}
```

It is used to navigate through arrays and collections. The following examples will demonstrate the difference b/w normal for loop and enhanced for loop.

Example #6: Read elements from single Dimensional Array using Basic for loop.

```
class Arrays {
	public static void main(String args[]) {
		int x[] = { 10, 20, 30, 40, 50, 60, 70, 80, 90 };
		for (int i = 0; i < x.length; i++)
			System.out.print(x[i] + "\t");
	}
}
```

Example #7: Read Single Dimensional Array elements using Enhanced for Loop.

```
class Arrays {
	public static void main(String args[]) {
		int x[] = { 10, 20, 30, 40, 50, 60, 70, 80, 90 };
		for (int a : x)
			System.out.println(a + "\t");
	}
}
```

(Use JDK 1.5 or later version to run it)

Example #8: Read Two dimensional array elements using Basic for loop.

```
class Arrays {
	public static void main(String args[]) {
		int x[][] = { { 10, 20, 30 }, { 40, 50, 60 }, { 70, 80, 90 } };
		for (int i = 0; i < 3; i++) {
			for (int j = 0; j < 3; j++)
				System.out.print(x[i][j] + "\t");
			System.out.println();
		}
	}
}
```

Example #9: Read Two Dimensional Array elements using Enhanced for loop.

```
class Arrays {
	public static void main(String args[]) {
		int x[][] = { { 10, 20, 30 }, { 40, 50, 60 }, { 70, 80, 90 } };
		for (int a[] : x) {
			for (int b : a)
				System.out.print(b + "\t");
			System.out.println();
		}
	}
}
```

(use later version to execute it)

Defining classes with data member and methods

Example #10: Student class with constructors

```
class Student {
        private int rollno;
        private String name, course;
        // constructor
        public Student() {
                rollno = 20;
                name = "";
                course = "";
        }
        // parameterized constructor
        public Student(int r, String nm, String C) {
                rollno = r;
                name = nm;
                course = C;
        }
        public void disp() {
                System.out.println("Roll no: " + rollno);
                System.out.println("Name: " + name);
                System.out.println("course:" + course);
        }
        public static void main(String args[]) {
                Student s = new Student();
                s.disp();
                Student s1 = new Student(102, "RAMAN", "J@2EE");
                s1.disp();
        }
}
```

Example # 11: Reading and printing Customer details:

```
 class customer {
        private int ID;
        private String name;
        // default constructor
        public customer() {
        }
        // parameterized constructor
        public customer(int ID, String name) {
                this.ID = ID;
                this.name = name;
        }
        public void display() {
                System.out.println("ID:" + ID);
                System.out.println("Name: " + name);
        }
}
```

```
class customerDemo {
	public static void main(String args[]) {
		customer c1 = new customer(11, "kiran");
		customer c2 = new customer(12, "ram");
		c1.display();
		c2.display();
	}
}
```

Example # 12: Calling methods through nameless Object

```
class Alpha {
	public Alpha() {
		System.out.println("Hello");
	}
	public void display() {
		System.out.println("world");
	}
}
class NamelessObject {
	public static void main(String args[]) {
		new Alpha().display();
	}
}
```

Example # 13: To find out free memory before and after garbage collections

```
class GarbageDemo {
	public static void main(String args[]) {
		int i;
		long a;
		Runtime r = Runtime.getRuntime();
		Long values[] = new Long[200];
		System.out.println("Amount of free Memory" + r.freeMemory());
		r.gc();
		System.out.println("Amount of free memory after creating array is"+ r.freeMemory());
		for (a = 10000, i = 0; i < 200; a++, i++) {
			values[i] = new Long(a);
		}
		System.out.println("amount of free Memory after creating array is:"+ r.freeMemory());
		for (i = 0; i < 200; i++) {
			values[i] = null;
		}
		System.out.println("amount of free Memory after creating array is:"+ r.freeMemory());

	}

}
```

Example # 14: Reading employee details and calculating net salary

```
import java.io.*;
class Emp {
        private int empno;
        private String name;
        private float salary, hra, da, pf, netsal;
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        public void read() throws IOException {
                System.out.println("Enter empono, name an salary");
                empno = Integer.parseInt(br.readLine());
                name = br.readLine();
                salary = Float.parseFloat(br.readLine());
        }
        public void calc() {
                hra = 0.1f * salary;
                da = 0.15f * salary;
                pf = 0.2f * salary;
                netsal = (salary + hra + da) - pf;
        }
        public void disp() {
                System.out.println("empno  : " + empno);
                System.out.println("Name  : " + name);
                System.out.println("Netsal  :" + netsal);
        }
}
class EmpDemo {
        public static void main(String args[]) throws IOException {
                Emp e = new Emp();
                e.read();
                e.calc();
                e.disp();
        }
}
```

Example # 15:An Enterprise bank application

```
import java.io.BufferedReader;
import java.io.DataInputStream;
import java.io.IOException;
import java.io.InputStreamReader;
class BankDemo {
        static BufferedReader br = new BufferedReader(new InputStreamReader(
                        System.in));
        int accNo, deposit, withdraw;
        Float bal;
        String accType, custName;
        public BankDemo() throws IOException {
                System.out.println("Enter account Number :");
                accNo = Integer.parseInt(br.readLine());
                System.out.println("Enter Initial Deposit: ");
```

```
            bal = Float.parseFloat(br.readLine());
            if (bal < 1000) {
                  System.out.println("Initial deposit should beat a least 1000");
                  System.exit(0);
            }
            System.out.println("Enter Account Type :");
            accType = br.readLine();
            System.out.println("Enter Customer Name:");
            custName = br.readLine();
}
public void deposit() throws IOException {
            System.out.println("Enter Amount to deposit :");
            deposit = Integer.parseInt(br.readLine());
            bal = bal + deposit;
            System.out.println("balance in account no. " + accNo + " is " + bal);
}
public void withdraw() throws IOException {
            DataInputStream din = new DataInputStream(System.in);
            System.out.println("Enter Amount to withdraw:");
            withdraw = Integer.parseInt(br.readLine());
            if (withdraw > (bal - 1000))
                  System.out.println("sorry! You can withdraw only rs. "
                              + (bal - 1000) + "/-");
            else {
                  bal = bal - withdraw;
                  System.out.println("balance in account no ." + accNo + "is" + bal);
            }
}
public void balanq() {
            System.out.println("Ledger Balance in Account No. " + accNo + "is"+ bal);
            System.out.println("Available Balance in Account no. " + accNo + "is"+ (bal - 1000));
}
public static void main(String args[]) throws IOException {
            int choice;
            BankDemo b = new BankDemo();
            do {
                  System.out.println("\t\tBANKING SYSTEM");
                  System.out.println("1.Deposit");
                  System.out.println("2. Withdraw");
                  System.out.println("3. Bal Enq");
                  System.out.println("4.Exit");
                  System.out.println("Enter your choice:");
                  choice = Integer.parseInt(br.readLine());
                  switch (choice) {
                  case 1:
                        b.deposit();
                        break;
                  case 2:
                        b.withdraw();
                        break;
```

```
                    case 3:
                            b.balanq();
                            break;
                    case 4:
                            System.exit(0);
                            break;
                    }
            } while (choice != 4);
    }
}
```

Example #16: Automation of bookings at a theatre

Case study

Booking process at a theatre needs to be computerized. The theatre has three type of tickets namely Diamond, and gold. Their and price is as follows:

Ticket category	Price	Seating Capacity
Diamonds	1000	200
Golds	500	250
Silvers	300	300

Develop an interaspire Java application, which contains sales and report modules. Sales module will keep track of sales, seats available and collection while the report module is responsible o print the sales details.

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
class Theatre {
        int no, cls;
        Boolean flag = false;
        final int dprice = 1000, gprice = 500, sprice = 300;
        int dSold, gSold, SSold, dRem, gRem, sRem;
        float dCol, gCol, sCol, totalCol, amt, amount, change;
        static BufferedReader br = new BufferedReader(new
        InputStreamReader(System.in));
        public Theatre() {
                dRem = 200;
                gRem = 250;
                sRem = 300;
                dCol = gCol = totalCol = 0.0F;
        }
        public void sales() throws IOException {
                System.out
                                .println("Enter class: \n\t1.Diamonds\n\t2.Golds\n\t3.Silvers");
                cls = Integer.parseInt(br.readLine());
                System.out.println("Enter no.of tickets: ");
                no = Integer.parseInt(br.readLine());
                switch (cls) {
                case 1:
```

```
                if (dRem >= no) {
                        amt = no * dprice;
                        dRem -= no;
                        dSold += no;
                        dCol += amt;
                        flag = true;
                } else {

                        System.out.println("Sorry!" + dRem + "diamonds are available");
                        br.readLine();
                }
                break;
        case 2:
                if (gRem >= no) {
                        amt = no * gprice;
                        dRem -= no;
                        dSold += no;
                        dCol += amt;
                        flag = true;
                } else {
                        System.out.println("Sorry! Only  " + gRem
                                        + "golds are available");
                        br.readLine();
                }
                break;
        case 3:
                if (sRem >= no) {
                        amt = no * sprice;
                        dRem -= no;
                        dSold += no;
                        dCol += amt;
                        flag = true;
                } else {
                        System.out.println("Sorry! Only " + sRem
                                        + "Silvrers  are available");
                        br.readLine();
                }
                break;
        }
        if (flag == true) {
                System.out.println("\n\tAmount is : " + amount);
                System.out.println("\n\tEnter amount received from customers: ");
                amount = Float.parseFloat(br.readLine());
                if (amount < amt) {
                        change = amt - amount;
                        System.out.println("Kindle collect Rs. " + change
                                        + "/-from customers");
                } else if (amount > amt) {
                        change = amount - amt;
                        System.out.println("Kindly return Rs. " + change
```

```
                                        + "/- to customers");
                } else
                        System.out.println("OK");
                flag = false;
            }
        }
        public void report() throws IOException {
            System.out.println("\n\t\tSALES REPORT");
            System.out.println("\n\t\tDiamonds sold: " + dSold);
            System.out.println("\tGolds sold:" + gSold);
            System.out.println("\tSilvers sold:" + SSold);
            System.out.println("\n\nDiamonds collections:" + dCol);
            System.out.println("\tGolds' Collection." + gCol);
            System.out.println("\tSilvers' Collections:" + sCol);
            totalCol = gCol + dCol + sCol;
            System.out.println("\n\tTotal Collections of the show:" + totalCol);
            System.out.println("\n\t\t***HOUSEFUL***");
            br.readLine();// waits for a key
        }
}
class ImaxTheatre {
        public static void main(String args[]) throws IOException {
            int choice;
            Theatre t = new Theatre();
            do {
                System.out.println("\n\t\tWELCOME TO BOOKINGS\n");
                System.out.println("Diamonds\tGolds\t\tSilvers");
                System.out.println( + t.dRem + "\t\t" + t.gRem + "\t\t" + t.sRem);
                System.out.println("\n\t1.Sales");
                System.out.println("\t2.Reports");
                System.out.println("\t3.Exit");
                System.out.println("\tEnter your choice:");
                choice = Integer.parseInt(t.br.readLine());
                switch (choice) {
                case 1:
                        t.sales();
                        break;
                case 2:
                        t.report();
                        break;
                case 3:
                        System.out.println("\n\n\tThen Q");
                        System.out.println("\tpress any key to exit…");
                        t.br.readLine();// waits for a key
                }
            } while (choice != 3);
        }
}
```

2.PACKAGES

Example #1: Creating user defined package

Problem:

Create apackage by name college and include classes such as Student and Teacher in the package.

Solution:

The following program defines a new packge by name college and keeps Student.class file in the package.

```
package college;
public class Student {
        private int rollno;
        private String sname;
        public Student(){}
        public Student(int rno, String name){
                rollno = rno;
                sname = name;
        }
        public void disp() {
                System.out.println("\nStudent Details");
                System.out.println("Roll number:" + rollno);
                System.out.println("Name: " + sname);
        }
}
```

Assume that your working directory is D:\Aspire
Save the above file in D:\Aspire

Compile it

D:\aspire> Java-d . Student.java

The above command creates a new directory (package) by name college in D:\Aspire and keeps Student.class file in D:\Aspire\college directory.

Making use of user defined package

```
import college.Student;
import java.io.*;
class StudDemo {
   public static void main (String args[]){
        int rno = Integer.parseInt(args[0]);
        String name = args[1];
        Student s=new Student(rno, name);
         s.disp();
   }
```

```
}
```
Save the above file in D:\Aspire
D:\aspire>javac StudDemo.java
D:\aspire>java StudDemo 101 Srikanth

Try It Out

Define a class by name Teacher with data members like techId, name, subject and methods such as getData()and put Data().Include this class in college package.

Example # 2: Creating a sub package

Problem: Create a sub package by name library in college package.Include Book.class and Member.class files into it.

Solution:

```
package college.library;
import java.io.*;
public class Member {
        private String memCode, memName;
        DataInputStream in = new DataInputStream(System.in);

        public void read() {
                try {
                        System.out.println("Enter meber code and name");
                        memCode = in.readLine();
                } catch (IOException e) {
                }
        }
        public void disp() {
                System.out.println(memCode + "\t" + memName);
        }
}
```

Assume that your working directory is D:\Aspire
Save the above file in D:\Aspire

Compilation:
javac-d D:\Aspire Member.java

This command creates a directory (sub package) in d:\Actice\college directory and keep Member.class file in it.

Making use of sub package

```
import college.library.Member;
import java.io.*;
class MemberDemo {
        public static void main(String args[]) throws IOException {
                Member m = new Member();
```

```
                m.read();
                m.disp();
        }
}
}
```

Save it d:\aspire

Compile and run it.

Try It Out

In the same manner define aclass by name Book with data members like bookId,title,author,publisher,price and methods like getData()and putData().Include its.class file into the sub package library.Then write a demo program to make use of it.

3.EXCEPTION HANDLING

Example #1: Demonstrates Arithmetic Exception

```
class Divide {
        public static void main(String args[]) {
                int a, b, res;
                a = Integer.parseInt(args[0]);
                b = Integer.parseInt(args[1]);
                try {
                        res = a / b;
                        System.out.println("Result: " + res);
                } catch (ArithmeticException e) {
                        System.out.println("Number can't be divided by zero");
                } finally {
                        System.out.println("Finally executed");
                }
        }
}
```

Execute the a bove program

Java Divide 10 2(gives output)

Java Divide 10 0(causes Arithmetic Exception)

Example #2: Demonstrates try, catch(es),generic catch and finally blocks

```
class ExceptDemo {
        public static void main(String args[]) {
                int a, b, res;
                try {
                        a = Integer.parseInt(args[0]);
                        b = Integer.parseInt(args[1]);
                        res = a / b;
                        System.out.println("Result is " + res);
                } catch (ArithmeticException ae) {
                        System.out.println("No.can't be devided by zero.pls enter a non zero value for b");
                } catch (ArrayIndexOutOfBoundsException e) {
                        System.out.println("Insufficient arguments.pls pass two integers");
                } catch (NumberFormatException e) {
                        System.out.println("Not a number");
                } catch (Exception e) {
                        System.out.println(e.getMessage());
                } finally {
                        System.out.println("Cleaning process goes here");
                }
        }
}
```

In the above program, the code kept in try block may cause the following exceptions.

1.Arithmetic Exception

2.Array Index Out Of Bounds Exception

3.Number Format Excepation and any unpredicted exception.

Execution the above program

Java ExcepDemo 10 2 (no exception, it gives result)

Java ExcepDemo10 0(Causes Arithmetic Exception)

Java ExcepDemo10 (causes Array Index Out Of Bounds Exception)

Java ExcepDemo10 ravi (causes NumberFormatException)

Example #3: Demonstrates Number Format Exception

```
class Square {
        public static void main(String[] args) {
                try {
                        int x = Integer.parseInt(args[0]);
                        System.out.println("Square is" + (x * x));
                } catch (NumberFormatException ne) {
                        ne.printStackTrace();
                }
        }
}
```

Execute the above program

Java Square 7 (gives output)
Java Square seven (Causes Number Format Exception)

Example #4: Execute the program given below, with demonstrates NegativeArray SizeException.Write down the the output.Study the flow of control in the program based on the output.

```
class Testing {
        public void divide() {
                try {
                        int a = 10, b = 0, c;
                        c = a / b;
                } catch (ArithmeticException e) {
                        System.out.println("Can't divide by zero");
                        doIt();
                        System.out.println("Hello");
                }
        }
        public void doIt() {
                try {
                        int x[] = new int[-10];
                } catch (NegativeArraySizeException e) {
                        System.out.println("Negative arry  size");
```

```
            } finally {
                System.out.println("from doIt() finally");
            }
        }
        public static void main(String args[]) {
            Testing t = new Testing();
            t.divide();
        }
}
```

Example # 5: Demonstrates nested try block

Execute the program given below with various kinds of command line arguments and observe the flow of control in the program.

```
class NestedTry {
    public static void main(String args[]) {
        int a, b, res;
        try {
            a = Integer.parseInt(args[0]);
            b = Integer.parseInt(args[1]);
            try {
                int x[] = new int[-10];
            } catch (NegativeArraySizeException ee) {
                System.out.println("Negative array size");
            } finally {
                System.out.println("Inner finally executed");
            }
            res = a / b;
            System.out.println("Result:" + res);
        } catch (ArithmeticException e) {
            System.out.println("Can't divide by zero");
        } catch (Exception e) {
            System.out.println(e.getMessage());
        } finally {
            System.out.println("Oouter finally executed");
        }
    }
}
```

Example # 6: Throwing exceptions manually using throw

```
class ThrowDemo {
    public static void main(String args[]) {
        int a = Integer.parseInt(args[0]);
        try {
            if (a == 0)
                throw new ArithmeticException();
```

```
                else
                        System.out.println("Its a valid Number");
            } catch (ArithmeticException e) {
                System.out.println("Number should not be zero");
            }
        }
}
```

Example # 7: Demonstrates throws clause used by divide() to by pass Arithemeic Exception to its caller

```
class ThrowsDemo1 {
    public static void main(String args[]) {
        int a, b;
        try {
            a = Integer.parseInt(args[0]);
            b = Integer.parseInt(args[1]);
            divide(a, b);
        } catch (ArithmeticException e) {
            System.out.println("Can't divide by zero");
        } catch (ArrayIndexOutOfBoundsException ae) {
            System.out.println("Insufficient arumments ");
        }
    }
    public static void divide(int x, int y) throws ArithmeticException {
        int c = x / y;
        System.out.println("Exception caught :" + c);
    }
}
```

Example # 8: Demonstrating generic catch, with can handle any kind of exception

```
// catching all types of exceptions
class CatchAllTypes {
    public static void main(String[] args) {
        int a, b, c;
        a = 10;
        b = 0;
        try {
            c = a / b;
            System.out.println("This will never Execute:");
        } catch (Exception ae) {
            System.out.println("Exception caught:" + ae);
        }
    }
}
```

Example # 9: Demonstrates user-defined exceptions

```
//User Defined unchecked exception class
class AgeException extends RuntimeException {
        public String toString() {
                return "Age should be b/w 25 and 35";
        }
}
class AgeExceptionDemo {
        public static void main(String args[]) {
                int a = Integer.parseInt(args[0]);
                if (a > 35 || a < 25)
                        throw new AgeException();
                else
                        System.out.println("Eligible");
        }
}
```

Observation

TheAge Exception class extends Runtime Exception class.Hence our exception is known as unchecked exception.Compile wil not verify it.There fore it is not mandatory to handle or declare it.

Execute the program

Java Age ExceptionDemo27 (prints Eligible)
Java Age ExceptionDemo10(causes Age Exception)

Try it Out

Rewrite the above program by extending Exception class.Then it becomes a checked exception.In this case case you will be forced to either handle it using try-catch or declaring it using throws.

4.MULTITHREADING

Example #1: Demonstrates controlling main thread

The program given below uses sleep() method to interrupt main thread. It also prints thread name, thread name, thread group and priority of thread.

```
class MainThread {
        public static void main(String args[]) throws InterruptedException {
                System.out.println("J2SE");
                Thread.sleep(2000);
                System.out.println("J2EE");
                Thread.sleep(2000);
                System.out.println("J2ME");
                System.out.println(Thread.currentThread());
                System.out.println(Thread.currentThread().getName());
        }
}
```

Example # 2: Demonstrates controlling main thread

In this example we change the name of main thread and print its details.

```
import java.lang.*;
class MainThread1 {
        public static void main(String[] args) {
                Thread t = Thread.currentThread();
                System.out.println("current Thread:" + t);
                t.setName("My Thread");
                System.out.println("After NameChange:" + t);
                try {
                        for (int i = 5; i > 0; i--) {
                                System.out.println(i);
                                Thread.sleep(1000);
                        }
                } catch (InterruptedException E) {
                        System.out.println("Main ThredInterrupted");
                }
        }
}
```

Example # 3: Creating multiple threads by extending Thread class

```
class Thread10 extends Thread {
        public void run() {
                System.out.println("Thread 1 starts");
                for (int i = 1; i < 10; i++) {
                        System.out.println("\t" + i);
                        try {
                                Thread.sleep(2000);
                        } catch (InterruptedException e) {
                        }
```

```
                }// for
                System.out.println("Thread 1 ends");
        }// run()
}// class
class Thread20 extends Thread {
        public void run() {
                System.out.println("Thread 2 starts");
                for (int i = 10; i >= 1; i--) {
                        System.out.println("\t" + i);
                        try {
                                Thread.sleep(2000);
                        } catch (InterruptedException e) {
                        }
                }// for
                System.out.println("Thread2 ends");
        }// run()
}// class
class ThreadDemo1 {
        public static void main(String args[]) throws InterruptedException {
                System.out.println("Main Thread starts");
                Thread10 t1 = new Thread10();
                Thread20 t2 = new Thread20();
                t1.start();
                t2.start();
                System.out.println("Main Thead ends");
        }
}
```

Observation

Two threads run simultaneously.Main thread starts first will not wait for the child threads to complete.

Do It YoureseIf

Modify the main() in the above program as follows and observe the output.

```
public class IntrruptedException {
        public static void main(String args[]) throws InterruptedException {
                System.out.println("Main Thread starts");
                Thread t1 = new Thread();
                Thread t2 = new Thread();
                t1.start();
                t2.start();
                t1.join();// main() thread waits for t1 to die
                t2.join();
                System.out.println("Main Thread ends");
        }
}
```

Observation

Main thread waits until the threads are finished.

Try It Out

Write a program, which creates three threads that prints words like J2EE,J2SE,JAVA respectively and in fintely.

Example # 4: Creating threads by implementing Runnable interface

```
class First implements Runnable {
        public void run() {
                try {
                        for (int i = 1; i <= 10; i++) {
                                System.out.println("\t" + i);
                                Thread.sleep(2000);
                        }// for
                }// try
                catch (InterruptedException e) {
                }
        }// run()
}// class
class Second implements Runnable {
        public void run() {
                try {
                        for (int i = 10; i >= 1; i--) {
                                System.out.println("\t" + i);
                                Thread.sleep(2000);
                        }// for
                }// try
                catch (InterruptedException e) {
                }
        }// run()
}// class
class ThreadsDemo {
        public static void main(String args[]) {
                First obj1 = new First();
                Second obj2 = new Second();
                Thread t1 = new Thread(obj1);
                Thread t2 = new Thread(obj2);
                t1.start();
                t2.start();
        }
}
```

Observation

In main(),as obj1 and obj2 are Runnable instances (they won't support thread's lifecycle mthods) they are assigned to thread instance as t1 and t2 respectively.

Example # 5: Creating multiple named threads by implementing Runnable interface

```
class Demo implements Runnable {
        Thread t;
        Demo(String threadname) {
                t = new Thread(this, threadname);
                System.out.println("New Thread:" + t);
                t.start();
        }
        public void run() {
                try {
                        for (int i = 5; i > 0; i--) {
                                System.out.println("Child Thread:" + i);
                                Thread.sleep(2000);
                        }
                } catch (InterruptedException ie) {
                        System.out.println("child intrrupted");
                }
                System.out.println("Exiting child thread" + t);
        }
}
public class MultipleThreads {
        public static void main(String args[]) {
                new Demo("One");
                new Demo("two");
                new Demo("three");
        }
}
```

Example # 6: Demonstrates the usage of join() and is Alive() methods

```
class Demo implements Runnable {
        Thread t;
        Demo(String threadname) {
                t = new Thread(this, threadname);
                System.out.println("New Thread:" + t);
                t.start();
        }
        public void run() {
                try {
                        for (int i = 5; i > 0; i--) {
                                System.out.println("Child Thread:" + i);
                                Thread.sleep(1500);
                        }
                } catch (InterruptedException e) {
                        System.out.println("Child interrupted");
                }
                System.out.println("Exiting child thread" + t);
        }
}
public class DemoOnIsAliveJoin {
```

```
        public static void main(String args[]) {
                Demo ob1 = new Demo("One");
                Demo ob2 = new Demo("Two");
                Demo ob3 = new Demo("Three");
                System.out.println("Thread one is Avlive :" + ob1.t.isAlive());
                System.out.println("Thread twois Avlive :" + ob2.t.isAlive());
                System.out.println("Thread Threeis Avlive :" + ob3.t.isAlive());
                System.out.println("Main thread Exiting");
        }
}
```

Example # 7: Demonstrates thread priorities

```
import java.lang.*;
class Demo implements Runnable {
        Thread t;
        Demo() {
                t = new Thread(this);
                t.start();
        }
        public void run() {
                try {
                        for (int i = 5; i > 0; i--) {
                                System.out.println("Child Thread:" + i);
                                Thread.sleep(500);
                        }
                } catch (InterruptedException e) {
                        System.out.println("Child Interrpted");
                }
                System.out.println("Exiting child thread" + t);
        }
}
public class ThreadPriorityDemo {
        public static void main(String args[]) {
                Demo ob1 = new Demo();
                Demo ob2 = new Demo();
                Demo ob3 = new Demo();
                ob1.t.setName("One");
                ob2.t.setName("Two");
                ob3.t.setName("Three");
                ob1.t.setPriority(3);
                ob2.t.setPriority(7);
                ob3.t.setPriority(Thread.MAX_PRIORITY);
                try {
                        for (int i = 5; i > 0; i--) {
                                System.out.println("Main Thread:" + i);
                                Thread.sleep(1000);
                        }
                } catch (InterruptedException e) {
                        System.out.println("Main Thread Interrupted");
```

```
                System.out
                        .println("Priority for Thread One=" + ob1.t.getPriority());
                System.out.println("Priority for Thread Two=" + ob2.t.getPriority());
                System.out.println("Priority for Thread Three="
                        + ob3.t.getPriority());
                System.out.println("Main Thread Exiting");
        }
    }
}
```

Example # 8: Thread Synchronizing using synchronized method

```
class Resource {
    synchronized public static void disp() {
        System.out.print("[");
        System.out.print("HELLO");
        try {
            Thread.sleep(1000);
        } catch (Exception e) {
        }
        System.out.println("]");
    }
}
class MyThread extends Thread {
    public void run() {
        Resource.disp();
    }
}
class Sync {
    public static void main(String args[]) {
        MyThread t1 = new MyThread();
        MyThread t2 = new MyThread();
        MyThread t3 = new MyThread();
        t1.start();
        t2.start();
        t3.start();
    }
}
```

Try It Out

Observe the output of the above program, without using synchronized keyword with disp()method.

Example # 9: Demonstrates thread synchronization using synchronized block

```
class CallMe {
    void call(String msg) {
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            System.out.println("Interrupted");
```

```
                }
                System.out.println("]");
        }
}
class Caller implements Runnable {
        String msg;
        CallMe target;
        Thread t;
        public Caller(CallMe targ, String s) {
                target = targ;
                msg = s;
                t = new Thread(this);
                t.start();
        }
        // Synchronize calls to call()
        public void run() {
                synchronized (target) {
                        target.call(msg);
                }
        }
}
class Synch {
        public static void main(String args[]) {
                CallMe target = new CallMe();
                Caller ob1 = new Caller(target, "Hello");
                Caller ob2 = new Caller(target, "Synchonized");
                Caller ob3 = new Caller(target, "World");
                // wait for threads to end
                try {
                        ob1.t.join();
                        ob2.t.join();
                        ob3.t.join();
                } catch (InterruptedException e) {
                        System.out.println("Interrupted");
                }
        }
}
```

Example # 10 : Demonstrates inter-thread communication using wait().notify() methods

```
class Q {
        int n;
        boolean valueset = false;
        synchronized int get() {
                if (!valueset) {
                        try {
                                wait();
                        } catch (Exception e) {
                                System.out.println("Interrupted Exception");
                        }
                }
```

```
                try {
                        System.out.println("Got:" + n);
                        Thread.sleep(2000);
                } catch (InterruptedException ee) {
                }
                valueset = false;
                notify();
                return n;
        }
        synchronized void put(int n) {
                if (valueset) {
                        try {
                                wait();
                        } catch (Exception e) {
                                System.out.println("Interrupted Exception");
                        }
                }
                this.n = n;
                valueset = true;
                try {
                        System.out.println("put:" + n);
                        Thread.sleep(2000);
                } catch (InterruptedException ee) {
                }
                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 ProducerConsumerDemo {
        public static void main(String[] args) {
                Q q = new Q();
                new Producer(q);
                new Consumer(q);
                System.out.println("press Control+ C to stop:");
        }
}
```

Example # 11: Working with thread groups

```
class NewThread extends Thread {
        boolean suspendFlag;

        NewThread(String threadname, ThreadGroup tgob) {
                super(tgob, threadname);
                System.out.println("New Thread:" + this);
                start();
        }
        public void run() {
                try {
                        for (int i = 5; i > 0; i--) {
                                System.out.println(getName() + ":" + i);
                                Thread.sleep(1000);
                                synchronized (this) {
                                        while (suspendFlag) {
                                                wait();
                                        }
                                }
                        }
                } catch (Exception e) {
                        System.out.println("Exception in" + getName());
                }
                System.out.println(getName() + "exiting");
        }
        void mysuspend() {
                suspendFlag = false;
        }
        synchronized void myresume() {
                suspendFlag = false;
                notify();
        }
}
class ThreadGroupDemo {
        public static void main(String[] args) {
                ThreadGroup groupA = new ThreadGroup("Group A");
                ThreadGroup groupB = new ThreadGroup("Group B");
                NewThread ob1 = new NewThread("One", groupA);
                NewThread ob2 = new NewThread("Two", groupA);
```

```
                NewThread ob3 = new NewThread("Three", groupB);
                NewThread ob4 = new NewThread("Four", groupB);
                System.out.println("\n here is output fron list():");
                groupA.list();
                groupB.list();
                System.out.println();
                System.out.println("Suspending Group A");
                Thread tga[] = new Thread[groupA.aspireCount()];
                groupA.enumerate(tga);
                for (int i = 0; i < tga.length; i++) {
                        ((NewThread) tga[i]).mysuspend();
                }
                try {
                        Thread.sleep(4000);
                } catch (Exception e) {
                        System.out.println("Main Thread Interrupted");
                }
                System.out.println("Resuming Group A");
                for (int i = 0; i < tga.length; i++) {
                        ((NewThread) tga[i]).myresume();
                }
                try {
                        System.out.println("Waiting for threads to finish:");
                        ob1.join();
                        ob2.join();
                        ob3.join();
                        ob4.join();
                } catch (Exception e) {
                        System.out.println("Exception in Main Thread");
                }
                System.out.println("Main Thread Exiting");
        }
}
```

Example # 12: Demonstrates deadlocks

Execute the program given below to find whether deadlock occurs.

```
  class A {
        B b;
        synchronized void a1() {
                System.out.println("From a1");
                b.b2();
        }
        synchronized void a2() {
                System.out.println("From a2");
        }
}
class B {
        A a;
```

```
        synchronized void b1() {
                System.out.println("From b1");
                a.a2();
        }
        synchronized void b2() {
                System.out.println("From  b2");
        }
};
class Thread1 extends Thread {
        A a;
        Thread1(A a) {
                this.a = a;
        }
        public void run() {
                for (int i = 0; i < 10; i++)
                        a.a1();
        }
};
class Thread2 extends Thread {
        B b;
        Thread2(B b) {
                this.b = b;
        }
        public void run() {
                for (int i = 0; i < 10; i++) {
                        b.b1();
                }
        }
};
class DeadLockDemo {
        public static void main(String[] args) {
                A a = new A();
                B b = new B();
                a.b = b;
                b.a = a;
                Thread1 t1 = new Thread1(a);
                Thread2 t2 = new Thread2(b);
                t1.start();
                t2.start();
                try {
                        t1.join();
                        t2.join();
                } catch (Exception e) {
                        e.printStackTrace();
                }
                System.out.println("Done")

}
}
```

5.IO STREAMS

Example # 1: Reading contents of given file and writing the same to standard output

```
import java.io.*;
class ReadFromFile {
	public static void main(String args[]) throws Exception {
		FileInputStream fis = new FileInputStream(args[0]);
		int ch;
		while ((ch = fis.read()) != -1)
			System.out.print((char) ch);
		fis.close();
	}
}
```

Execution

Java ReeadFromFile sample.txt

O/P:

This is aspire technologies.

Example #2: Reading characters from user and writing them to a file

```
import java.io.*;
class WriteToFile {
	public static void main(String args[]) throws Exception {
		int ch;
		FileOutputStream fos = new FileOutputStream("sample.txt");
		BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
		System.out.println("Enter characters [@ to quit]");
		while ((ch = br.read()) != '@')
			fos.write((char) ch);
		fos.close();
	}
}
```

Example # 3: Accepting two filename through command line and copying contents of the first file to second file.

```
import java.io.*;
class Copy {
	public static void main(String args[]) throws IOException {
		FileInputStream fis = new FileInputStream(args[0]);
		FileOutputStream fos = new FileOutputStream(args[1]);
		int ch;
		while ((ch = fis.read())!= -1) {
			fos.write((char)ch);
		}
		fis.close();
```

```
                fos.close();
        }
}
```

Execution

Java copy sourcefile destfile

Example # 4:Reading and writing primitives

```
import java.io.*;
class ReadWritePrim {
        public static void main(String args[]) throws IOException {
                FileOutputStream fos = new FileOutputStream("primitive.txt");
                DataOutputStream dos = new DataOutputStream(fos);
                dos.writeInt(102);
                dos.writeBoolean(true);
                dos.writeChar('X');
                dos.writeFloat(123.3f);
                dos.close();
                fos.close();
                FileInputStream Fis = new FileInputStream("primitive.txt");
                DataInputStream dis = new DataInputStream(Fis);
                System.out.println(dis.readInt());
                System.out.println(dis.readBoolean());
                System.out.println(dis.readChar());
                System.out.println(dis.readFloat());
                Fis.close();
                dis.close();
        }
}
```

Example # 5: Writing strings to a file

```
import java.io.*;
class WriteStrings {
        public static void main(String args[]) throws IOException {
                FileOutputStream fos = new FileOutputStream("strings.txt");
                BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
                String str;
                byte b[];
                System.out.println("Enter characters CR to quit");
                while (!(str = br.readLine()).equals("")) {
                        b = str.getBytes();
                        fos.write(b);
                }
                fos.close();
        }
}
```

CR: Carriage return (enter key)

Example # 6:Reading from byte array

```
import java.io.*;
class ReadByteArray {
        public static void main(String args[]) throws IOException {
                String s = "abcdefghijklmnopqrstuvwxyz";
                byte b[] = s.getBytes();
                ByteArrayInputStream bis = new ByteArrayInputStream(b);
                int ch;
                while ((ch = bis.read()) != -1) {
                        System.out.println((char) ch);
                }
                bis.close();
        }
}
```

Example # 7: Working with Pushback input stream

```
import java.io.*;
class PushbackDemo {
        public static void main(String args[]) throws IOException {
                String s = "if (a==4)a=1";
                byte b[] = s.getBytes();
                ByteArrayInputStream bis = new ByteArrayInputStream(b);
                PushbackInputStream pis = new PushbackInputStream(bis);
                int ch;
                while ((ch = pis.read()) != -1) {
                        switch (ch) {
                        case '=':
                                if ((ch = pis.read()) =='=')
                                        System.out.print(".eq.");
                                else {
                                        System.out.print(".eq.");
                                        pis.unread(1);
                                }
                                break;
                        default:
                                System.out.print((char)ch);
                                break;
                        }
                }
                System.out.println((char)ch);
        }
}
```

Example # 8: Appending data to an existing file using Random access File class

```
import java.io.*;
class RandomAccess {
        public static void main(String args[]) throws IOException {
                String source, dest;
                BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
                System.out.println("Enter source filename: ");
                source = br.readLine();
                System.out.println("Enter destination file:");
                dest = br.readLine();
                FileInputStream fis = new FileInputStream(source);
                RandomAccessFile r = new RandomAccessFile(dest, "rw");
                int ch;
                r.seek(r.length());
                while ((ch = fis.read()) != -1) {
                        r.write(ch);
                }
                System.out.println("Data trans fer complete");
                System.out.println("Press any key to exit");
                br.readLine();
                br.close();
                r.close();
        }
}
```

Example # 9: Combining streams using Sequence Input Stream Class

```
import java.io.*;
class SequnceInput {
        public static void main(String args[]) throws IOException {
                FileInputStream f1, f2;
                f1 = new FileInputStream(args[0]);
                f2 = new FileInputStream(args[1]);
                SequenceInputStream s;
                s = new SequenceInputStream(f1, f2);
                int ch;
                while ((ch = s.read()) != -1)
                    System.out.print((char) ch);
                f1.close();
                f2.close();
                s.close();
        }
}
```

Execution

```
Java SequenceInput file 1 file2
```

Example # 10: Object serialization and de-serialization

```
import java.io.*;
class Student implements Serializable {
        private int rollno;
        private String name, course;
        public void read() throws IOException {
                BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
                System.out.println("Enter rollno,name and course:");
                rollno = Integer.parseInt(br.readLine());
                name = br.readLine();
                course = br.readLine();
        }
        public void disp() {
                System.out.println("\t" + rollno + name + "\t" + course);
        }
}
class serializeDemo {
        public static void main(String args[]) throws Exception {
                // serialization
                FileOutputStream fos = new FileOutputStream("student.ser");
                ObjectOutputStream os = new ObjectOutputStream(fos);
                Student s = new Student();
                s.read();
                os.writeObject(s);
                fos.close();
                os.close();
                // Deserialization
                FileInputStream fis = new FileInputStream("Stdent.ser");
                ObjectInputStream is = new ObjectInputStream(fis);
                Student s1 = (Student) is.readObject();
                System.out.println("\n\tSTUDENT DETAILS");
                s1.disp();
                fis.close();
                is.close();
        }
}
```

Example # 11: Object serialization and de-serialization with transient data members

```
import java.io.*;
class Emp implements Serializable {
        private int empno;
        String name;
        private float sal;
        transient private float hra, da, pf, netsal;
```

```
	public void read() throws IOException {
		BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
		System.out.println("Enter empno,name and salary");
		empno = Integer.parseInt(br.readLine());
		name = br.readLine();
		sal = Float.parseFloat(br.readLine());
	}
	public void calc() {
		hra = 0.2f * sal;
		da = 0.1f * sal;
		pf = 0.15f * sal;
		netsal = (sal + hra + da) - pf;
	}
	public void disp() {
		System.out.println("Empno :" + empno);
		System.out.println("Name : " + name);
		System.out.println("Salary :" + sal);
		System.out.println("HRA : " + hra);
		System.out.println("DA :" + da);
		System.out.println("PF :" + pf);
		System.out.println("Netsal :" + netsal);
	}
}
class TransientDemo {
	public static void main(String args[]) throws IOException,
			ClassNotFoundException {
		Emp e = new Emp();
		e.read();
		e.calc();
		// before sending to file
		System.out.println("Before sending to file:");
		e.disp();
		FileOutputStream fos = new FileOutputStream("emp.txt");
		ObjectOutputStream os = new ObjectOutputStream(fos);
		os.writeObject(e);
		fos.close();
		os.close();
		// reading from file
		FileInputStream fis = new FileInputStream("emp.txt");
		ObjectInputStream is = new ObjectInputStream(fis);
		Emp e1 = (Emp) is.readObject();
		e1.disp();
		fis.close();
		is.close();
	}
}
```

Observation

In the out put 0.0is printed for hra,da,pf and netsal. This is because they are declared transient. Transient data member will not be part of persistent object.

Example # 12: Reading from file using character stream class File Reader[Not complied]

```
import java.io.*;
class FileWriteDemo {
        public static void main(String args[]) throws IOException {
                BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
                String file, path;
                System.out.println("Enter file name:");
                file = br.readLine();
                System.out.println("Enter path:");
                path = br.readLine();
                FileReader fr = new FileReader(path+ file);
                int ch;
                System.out.println("Contents of " + file);
                while ((ch = fr.read()) != -1) {
                        System.out.print((char) ch);
                }
                br.close();
                fr.close();
        }
}
```

Example # 13: Writing data to file using character stream class File Writer

```
import java.io.*;
class FileWriteDemo {
        public static void main(String[] args) {
                try {
                        FileWriter fw = new FileWriter(args[0]);
                        for (int i = 0; i < 12; i++) {
                                fw.write("line" + i + "\n");
                        }
                        fw.close();
                } catch (Exception e) {
                        System.out.println("Exception" + e);
                }
        }
}
```

6.WORKING WITH FILE SYSTEM

The following exercises demonstrate various operations on file system of Operating System.

Example # 1: Demonstrates the methods of File class

```
import java.io.*;
class FileDemo {
        public static void main(String args[]) {
                File f = new File("d:/j2se/io/Test.java");
                System.out.println("File Name:" + f.getName());
                System.out.println("It is a file:" + f.isFile());
                System.out.println("It is a directory:" + f.isFile());
                System.out.println("Abs path: " + f.getAbsolutePath());
                System.out.println("Hidden file: " + f.isHidden());
                System.out.println("File exists:" + f.exists());
                System.out.println("Parent " + f.getParent());
        }
}
```

Observation

In the above program, give the path of an existing file and study the output.

Example # 2: Creates a folder by name INDIA in D:

```
import java.io.*;
class CreateFolder {
        public static void main(String args[]) {
                File f = new File("d:/india");
                if (f.mkdir())
                        System.out.println("Folder 'INDIA'created");
                else
                        System.out.println("'Can't create the folder");
        }
}
```

Example # 3: Creates the given directory structure in D:

```
import java.io.*;
class CreateDirs {
        public static void main(String args[]) {
                File f = new File("d:/ABC/XYZ/HYD");
                if (f.mkdirs())
                        System.out.println("Dirs created");
                else
                        System.out.println("Dirs not created");
        }
}
```

Example # 4: Displays files and folders of D:

```
import java.io.*;
class DispFiles {
        public static void main(String args[]) {
                File f = new File("D:/");
                String files[] = f.list();
                for (int i = 0; i < files.length; i++) {
                        System.out.println(files[i]);
                }
        }
}
```

Example # 5: Renaming an existing file

```
import java.io.*;
class RenameFile {
        public static void main(String args[]) throws Exception {
                String oldfile, newfile;
                BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
                System.out.println("Enter existing file: ");
                oldfile = br.readLine();
                System.out.println("Enter new file name:");
                newfile = br.readLine();
                File f1 = new File(oldfile);
                File f2 = new File(newfile);
                f1.renameTo(f2);
                br.close();
        }
}
```

Example # 6: Searching for given file/folder in E:

```
import java.io.*;
class SearchFile {
        public static void main(String args[]) throws IOException {
                String file;
                DataInputStream in = new DataInputStream(System.in);
                System.out.println("Enter file/folder name");
                file = in.readLine();
                File f = new File("E:/");
                String files[] = f.list();
                boolean flag = false;
                for (int i = 0; i < files.length; i++) {
                        if (files[i].equals(file)) {
                                flag = true;
                                break;
                        }
                }
                if (flag)
```

```
                System.out.println("File/Folder exists");
            else
                System.out.println("File/Folder doesn.t exist");
        }
}
```

Example # 7: Prints the root directories available in the system

```
import java.io.File;
class ListRoots {
        public static void main(String args[]) throws Exception {
            File f = new File("d:/");
            File r[] = f.listRoots();
            for (int i = 0; i < r.length; i++)
                System.out.println(r[i]);
        }
}
```

Example # 8: Creates a new blank file the given directory

```
import java.io.*;
public class CreateFile {
        public static void main(String args[]) throws Exception {
            File f = new File("d:/ABC/india.txt");
            if (f.createNewFile())
                System.out.println("File created");
            else
                System.out.println("File is not created");
        }
}
```

Example # 9: Deletes the file given through command line

```
import java.io.*;
class DeleteFile {
        public static void main(String args[]) throws Exception {
            File f = new File(args[0]);
            if (f.delete())
                System.out.println("File deleted");
            else
                System.out.println("NO such file");

        }
}
```

7.NET WORKING

Exercise#1: Creating URL object and testing methods supported by URL class

```
import java.net.*;

class URLDemo {
	public static void main(String args[]) throws MalformedURLException {
		URL hp = new URL("http://www.yahoo.com");
		System.out.println("Protocol:" + hp.getProtocol());
		System.out.println("Port:" + hp.getPort());
		System.out.println("Host:" + hp.getHost());
		System.out.println("File:" + hp.getFile());
		System.out.println("Ext:" + hp.toExternalForm());
	}
}
```

Exercise#2:Creating Aspire Address object and testing its methods

```
//Demonstrate Aspire Address

import java.net.*;

class AspireAddressTest {
	public static void main(String args[]) throws UnknownHostException {
		AspireAddress address =AspireAddress.getLocalHost();
		System.out.println(address);
		System.out.println(address.getHostName());
		System.out.println(address.getHostAddress());
		System.out.println(address.getAddress());
		AspireAddress SW[] = AspireAddress.getAllByName("class");
		for (int i = 0; i < SW.length; i++)
			System.out.println(SW[i]);
	}
}
```

Exercise#3:Accessing data from alocal/remote system

```
import java.net.*;
import java.io.*;
class UrlConnection {
	public static void main(String[] args) throws Exception {
		URL u = new URL("http://sys2:8080/online/index.html");
		// or URL u=newURL (file:///d:/aspire/sample.txt);// to read from local
		// system
		URLConnection con = u.openConnection();
		con.setDoInput(true);
		con.setDoOutput(true);
```

```
                System.out.println("Connected..");
                InputStreamReader isr = new InputStreamReader(con.getInputStream());
                BufferedReader br = new BufferedReader(isr);
                FileOutputStream fos = new FileOutputStream("game.java");
                while (true) {
                        int i = br.read();
                        fos.write(i);
                        if (i == -1)
                                break;
                }
        }
}
```

Exercise#4:AC/S chat application

Compile and execute the following programs in two different machines.In client program,specify the name of computer to which the client connects.

```
//Chat Server

import java.io.*;
import java.net.*;
class Server {
        public static void main(String args[]) throws IOException {
                ServerSocket s = new ServerSocket(9080);
                Socket socket = null;
                BufferedReader br = null;
                BufferedReader in = null;
                PrintWriter out = null;
                System.out.println("Running:" + s);
                System.out.println("Waiting for clients:");
                try {
                        socket = s.accept();
                        br = new BufferedReader(new InputStreamReader(System.in));
                        in = new BufferedReader(new InputStreamReader(socket
                                        .getInputStream()));
                        out = new PrintWriter(new BufferedWriter(new OutputStreamWriter(
                                        socket.getOutputStream())), true);// Print Writer flushes

        // automatically
                        while (true) {
                                String str = in.readLine();
                                if (str.equals("END"))
                                        break;
                                System.out.println("From client:" + str);
                                System.out.println("Enter response:");
                                str = br.readLine();
                                out.println(str);
                        }
```

```
                } finally {
                        socket.close();
                        s.close();
                        br.close();
                        in.close();
                        out.close();
                }
        }
}

//Chat client

import java.io.*;
import java.net.*;
class Client {
        public static void main(String args[]) throws IOException {
                Socket s = new Socket("localhost", 9080);
                BufferedReader br = null;
                BufferedReader in = null;
                PrintWriter out = null;
                System.out.println("Running:" + s);
                try {
                        br = new BufferedReader(new InputStreamReader(System.in));
                        in = new BufferedReader(new InputStreamReader(s.getInputStream()));
                        out = new PrintWriter(new BufferedWriter(new OutputStreamWriter(s
                                        .getOutputStream())), true);
                        // Print Writer flushes automatically
                        while (true) {
                                System.out.println("Enter request:");
                                String str = br.readLine();
                                out.println(str);
                                str = in.readLine();
                                if (str.equals("END"))
                                        break;
                                System.out.println("From server:" + str);
                        }
                } finally {
                        s.close();
                        br.close();
                        in.close();
                        out.close();
                }
        }
}
```

Exercise#5:C/S application with FTPServer and FTPClient

```
//FTP Server

import java.net.*;
```

```
import java.io.*;
class FTPServer {
	public static void main(String[] args) throws Exception {
		ServerSocket server = new ServerSocket(8000);
		Socket s = server.accept();
		DataInputStream fromclient = new DataInputStream(s.getInputStream());
		String filename = fromclient.readLine();
		DataInputStream fromfile = new DataInputStream(new FileInputStream(
				filename));
		PrintStream toclient = new PrintStream(s.getOutputStream());
		String contents = fromfile.readLine();
		while (contents != null) {
			toclient.println(contents);
			contents = fromfile.readLine();
		}
		fromclient.close();
		toclient.close();
		fromfile.close();
		s.close();
	}
}

//FTPClient

import java.net.*;
import java.io.*;
class FTPClient {
	public static void main(String[] args) throws Exception {
		Socket client = new Socket("localhost", 8000);
		DataInputStream stdin = new DataInputStream(System.in);
		System.out.println("Enter the filename:");
		String filename = stdin.readLine();
		System.out.println();
		PrintStream toserver = new PrintStream(client.getOutputStream());
		toserver.println(filename);
		DataInputStream fromserver = new DataInputStream(client
				.getInputStream());
		String msg = fromserver.readLine();
		System.out.println("Contents sent from server…….");
		System.out.println();
		while (msg != null) {
			System.out.println(msg);
			msg = fromserver.readLine();
		}
		fromserver.close();
		toserver.close();
		stdin.close();
```

```
                client.close();
        }
}
```

Exrcise#6:C/Sapplication with UDP protocol(CUI)

```
//SERVER PROGRAM

import java.net.*;
import java.io.*;
public class DatagramServer {
        public static void main(String args[]) throws Exception {
                System.out.println("CRATING THE SERVER SOCKET");
                DatagramPacket rp, sp;
                // Create the Datagram Packet(new byte [512],512);
                rp = new DatagramPacket(new byte[512], 512);
                // Creat the datagramParam socket to establish connection
                DatagramSocket ds = new DatagramSocket(7777);
                System.out.println("DONE");
                while (true) {
                        System.out.println("WAITING FOR THE packet");
                        // Receive Datagram packet
                        ds.receive(rp);
                        System.out.println("GOT THE CLIENTpacket");
                        byte[] b1 = rp.getData();
                        int l = rp.getLength();
                        String str = new String(b1, 0, 1);
                        StringBuffer sb = new StringBuffer(str);
                        sb.reverse();
                        String msg = sb.substring(0) + " - - - - - - " + str;
                        byte[] b = msg.getBytes();
                        // create the datagram Packet to send…
                        sp = new DatagramPacket(b, b.length, rp.getAddress(), rp.getPort());
                        // send the datagram packet to client.
                        ds.send(sp);
                }
        }
}

//CLIENT PROGRAM

import java.net.*;
import java.io.*;
public class DatagramClient {
        public static void main(String args[]) throws Exception {
                // creating a datagram socket
                DatagramSocket ds = new DatagramSocket();
                DatagramPacket rp, sp;
                // to get data from the keyboard
```

```
                BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
                String s = br.readLine();
                byte arr[] = s.getBytes();
                // getting host address
                InetAddress inet = InetAddress.getLocalHost();
                // creating a DatagramPacket to send
                sp = new DatagramPacket(arr, arr.length, inet, 7777);
                System.out.println("The string sending…." + s + '\n');
                // sends the paket to server using datagram socket.
                ds.send(sp);
                // creating a Datagram packet to recivee
                rp = new DatagramPacket(new byte[512], 512);
                // receiving the packet
                ds.receive(rp);
                // printing the data onto the screen
                System.out.println(rp.getData());
        }
}
```

Exercies#7:C/S application with UDP protocol(GUI)

```
//Server program

import java.io.*;
import java.net.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class DatagramServer1 extends JFrame {
        JTextArea display;
        DatagramPacket sendPacket, receivePacket;
        DatagramSocket socket;

        public DatagramServer1() {
                super("Server");
                display = new JTextArea();
                getContentPane().add(new JScrollPane(display), BorderLayout.CENTER);
                setSize(400, 300);
                show();
                try {
                        socket = new DatagramSocket(5000);
                } catch (SocketException se) {
                        se.printStackTrace();
                        System.exit(1);
                }
        }
        public void waitForPackets() {
                while (true) {
                        try {
```

```
                        byte data[] = new byte[100];
                        receivePacket = new DatagramPacket(data, data.length);
                        // wait for packet
                        socket.receive(receivePacket);
                        // process packet
                        display.append("\nPacket received:"
                                + "\nFrom host:"
                                + receivePacket.getAddress()
                                + "\nHost port:"
                                + receivePacket.getPort()
                                + "\nLength:"
                                + receivePacket.getLength()
                                + "/ncontaining:\n\t"
                                + new String(receivePacket.getData(), 0, receivePacket
                                        .getLength()));
                        display.append("\n\nEcho data to cient…");
                        sendPacket = new DatagramPacket(receivePacket.getData(),
                                receivePacket.getLength(), receivePacket.getAddress(),
                                receivePacket.getPort());
                        socket.send(sendPacket);
                        display.append("packet sent\n");
                        display.setCaretPosition(display.getText().length());
                } catch (IOException io) {
                        display.append(io.toString() + "\n");
                        io.printStackTrace();
                }
        }
    }
    public static void main(String a[]) {
        DatagramServer1 app = new DatagramServer1();
        app.waitForPackets();
    }
}

//Client Proram

import java.io.*;
import java.net.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class DatagramClientGUI extends JFrame implements ActionListener {
    JTextField enter;
    JTextArea display;
    DatagramPacket sendPacket, receivePacket;
    DatagramSocket socket;
    public DatagramClientGUI() {
        super("Client");
        enter = new JTextField("Type message here");
        enter.addActionListener(this);
```

```
                getContentPane().add(enter, BorderLayout.NORTH);
                display = new JTextArea();
                getContentPane().add(new JScrollPane(display), BorderLayout.CENTER);
                setSize(400, 300);
                show();
                try {
                        socket = new DatagramSocket();
                } catch (SocketException se) {
                        se.printStackTrace();
                        System.exit(1);
                }
        }
        public void actionPerformed(ActionEvent e) {
                try {
                        display.append("\nSending paket containing" + e.getActionCommand()
                                        + "\n");
                        String s = e.getActionCommand();
                        byte data[] = s.getBytes();
                        sendPacket = new DatagramPacket(data, data.length,
                                        InetAddress.getLocalHost(), 5000);
                        socket.send(sendPacket);
                        display.append("Packet sent\n");
                        display.setCaretPosition(display.getText().length());
                } catch (IOException i) {
                }
        }
        public static void main(String a[]) {
                DatagramClientGUI app = new DatagramClientGUI();
        }
}
```

8.AWT

Exercise1:Creating a simple window

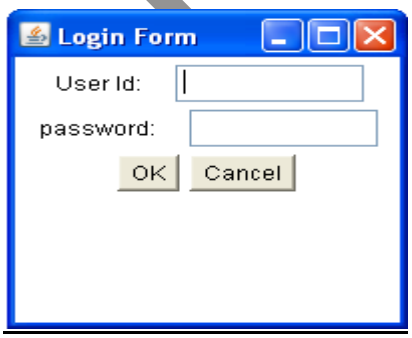
```
import java.awt.*;
class MyWindow extends Frame {
        public MyWindow() {
                setTitle("my First Window");
                setBackground(Color.orange);
                setSize(300, 300);
                setLocation(100, 100);
                setVisible(true);
        }

        public static void main(String args[]) {
                new MyWindow();
        }
}
```

Exercise2:Creating a simple window and drawing oval

```
import java.awt.*;
class MyWindow extends Frame {
        public MyWindow() {
                setTitle("My First Window");
                setBackground(Color.orange);
                setSize(300, 300);
                setLocation(100, 100);
                setVisible(true);
        }
        public static void main(String args[]) {
                new MyWindow();
        }
        public void paint(Graphics g) {
                g.setColor(Color.red);
                g.fillOval(100, 100, 100, 50);
        }
}
```

Exercise3:Designing the following user interface



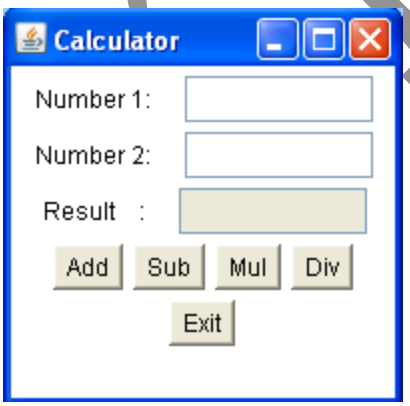
```
import java.awt.*;
class Login extends Frame {
	Label l1, l2;
	TextField t1, t2;
	Button b1, b2;
	Panel p1, p2;
	public Login() {
		setTitle("Login Form");
		p1 = new Panel();
		p1.setLayout(new GridLayout(2, 2));
		p2 = new Panel();
		l1 = new Label("User Id:");
		l2 = new Label("Password:");
		t1 = new TextField(10);
		t2 = new TextField(10);
		t2.setEchoChar('*');
		b1 = new Button("OK");
		b2 = new Button("CANCEL");
		p1.add(l1);
		p1.add(t1);
		p1.add(l2);
		p1.add(t2);
		p2.add(b1);
		p2.add(b2);
		add("North", p1);
		add("South", p2);
		validate();
		setSize(200, 200);
		setVisible(true);
	}
	public static void main(String args[]) {
		Login l = new Login();
	}
}
```

Exercise 4: Login Form with functionality

```
import java.awt.*;
import java.awt.event.*;
class Login extends Frame implements ActionListener {
	// declaring controls
	Label l1, l2;
	TextField t1, t2;
	Button b1, b2;
	// constructor
	public Login() {
		setTitle("Login Form");
		// Creating controls
		l1 = new Label("User Id:");
		l2 = new Label("password:");
		t1 = new TextField(10);
```

```
                t2 = new TextField(10);
                t2.setEchoChar('*');
                b1 = new Button("OK");
                b2 = new Button("Cancel");
                // registering controls with listener
                b1.addActionListener(this);
                b2.addActionListener(this);
                // setting flow layout
                setLayout(new FlowLayout());
                // adding controls to frame
                add(l1);
                add(t1);
                add(l2);
                add(t2);
                add(b1);
                add(b2);
                setSize(200, 200);
                setVisible(true);
        }
        public static void main(String args[]) {
                new Login();
        }
        public void actionPerformed(ActionEvent e) {
                if (e.getSource() == b1) {
                        if (t1.getText().equals("active") && t2.getText().equals("hyd"))
                                javax.swing.JOptionPane.showMessageDialog(this, "Valid User");
                        else
                                javax.swing.JOptionPane.showMessageDialog(this, "Invalid User");
                } else
                        System.exit(0);
        }// actionPerformed()
}// class
```

Exercise 5:Developing the following application



```
import java.awt.*;
import java.awt.event.*;
class Calc extends Frame implements ActionListener {
        Label l1, l2, l3;
        TextField t1, t2, t3;
        Button b1, b2, b3, b4, b5;
        float x, y, res = 0;
        public Calc(String title) {
                super(title);
                setLayout(new FlowLayout());
                l1 = new Label("Number 1:");
                l2 = new Label("Number 2:");
                l3 = new Label("Result   :");
                t1 = new TextField(10);
                t2 = new TextField(10);
                t3 = new TextField(10);
                t3.setEditable(false);
                b1 = new Button("Add");
                b2 = new Button("Sub");
                b3 = new Button("Mul");
                b4 = new Button("Div");
                b5 = new Button("Exit");
                b1.addActionListener(this);
                b2.addActionListener(this);
                b3.addActionListener(this);
                b4.addActionListener(this);
                b5.addActionListener(this);
                add(l1);
                add(t1);
                add(l2);
                add(t2);
                add(l3);
                add(t3);
                add(b1);
                add(b2);
                add(b3);
                add(b4);
                add(b5);
        }
        public static void main(String args[]) {
                Calc c = new Calc("Calculator");
                c.setSize(200, 200);
                c.show();
        }
        public void actionPerformed(ActionEvent e) {
                x = Float.parseFloat(t1.getText());
                y = Float.parseFloat(t2.getText());
                if (e.getSource() == b1)
                        res = x + y;
                else if (e.getSource() == b2)
```

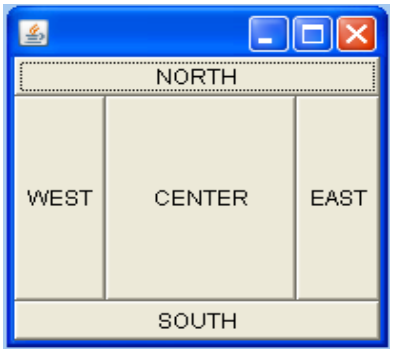
```
                res = x - y;
            else if (e.getSource() == b3)
                res = x * y;
            else if (e.getSource() == b4)
                res = x / y;
            else if (e.getSource() == b5)
                System.exit(0);
            t3.setText(String.valueOf(res));
        }
}
```

Exercise 6:Designing the following user interface

It demonstrates the usage of GridLayout

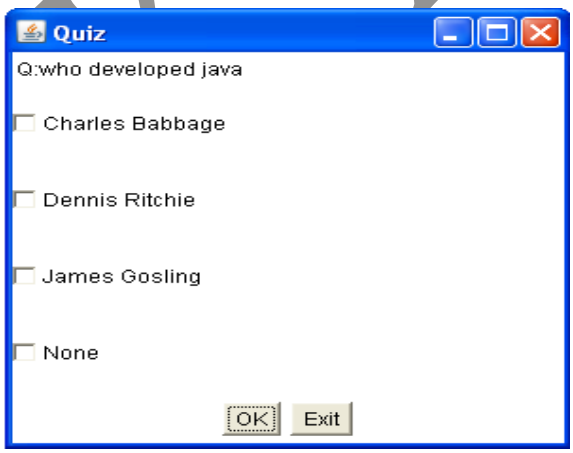

```
import java.awt.*;
class Grid extends Frame {
        Button b[];
        public Grid() {
            setTitle("Grid");
            // declaring button array
            b = new Button[16];
            // creating and adding buttons
            setLayout(new GridLayout(4, 4));
            for (int i = 0; i < 16; i++) {
                b[i] = new Button(" " + (i + 1));
                add(b[i]);
            }
            setBackground(Color.cyan);
            setSize(200, 200);
            setVisible(true);
        }
        public static void main(String args[]) {
            new Grid();
        }
}
```

Exercise 7: Demonstrates border layout




```
import java.awt.*;
class Buttons extends Frame {
        Button b1, b2, b3, b4, b5;
        public Buttons() {
                b1 = new Button("NORTH");
                b2 = new Button("SOUTH");
                b3 = new Button("EAST");
                b4 = new Button("WEST");
                b5 = new Button("CENTER");
                add("North", b1);
                add("South", b2);
                add("East", b3);
                add("West", b4);
                add("Center", b5);
                setSize(200, 200);
                setLocation(150, 150);
                setVisible(true);
        }
        public static void main(String args[]) {
                new Buttons();
        }
}
```

Exercise 8: Develop The following application



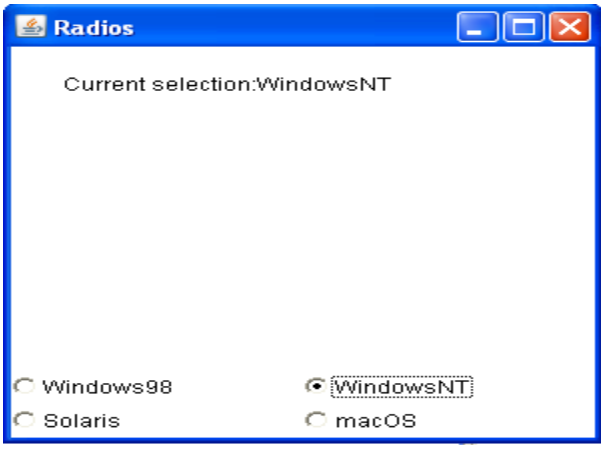
```
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
class Quiz extends Frame implements ActionListener {
        Label l1;
        Checkbox c1, c2, c3, c4;
        Button b1, b2;
        Panel p1, p2;
        public Quiz(String title) {
                super(title);
                // orset title(title);
                l1 = new Label("Q:who developed java");
                c1 = new Checkbox("Charles Babbage");
                c2 = new Checkbox("Dennis Ritchie");
                c3 = new Checkbox("James Gosling");
                c4 = new Checkbox("None");
                b1 = new Button("OK");
                b2 = new Button("Exit");
                b1.addActionListener(this);
                b2.addActionListener(this);
                p1 = new Panel();
                p1.setLayout(new GridLayout(4, 1));
                p1.add(c1);
                p1.add(c2);
                p1.add(c3);
                p1.add(c4);
                p2 = new Panel();
                p2.add(b1);
                p2.add(b2);
                add("North", l1);
                add("South", p2);
                add("West", p1);
                setSize(300, 300);
                setVisible(true);
        }
        public void actionPerformed(ActionEvent e) {
                if (e.getSource() == b1)
                        if (c1.getState() == false && c2.getState() == false
                                        && c3.getState() == true && c4.getState() == false)
                                javax.swing.JOptionPane.showMessageDialog(this,
                                                "congratulations!");
                        else
                                javax.swing.JOptionPane.showMessageDialog(this,
                                                "Sorry!Try Again");
                else
                        System.exit(0);
        }
        public static void main(String args[]) {
                new Quiz("Quiz");
        }
```

}

Do It Yourself

Try out the above application using radio buttons.

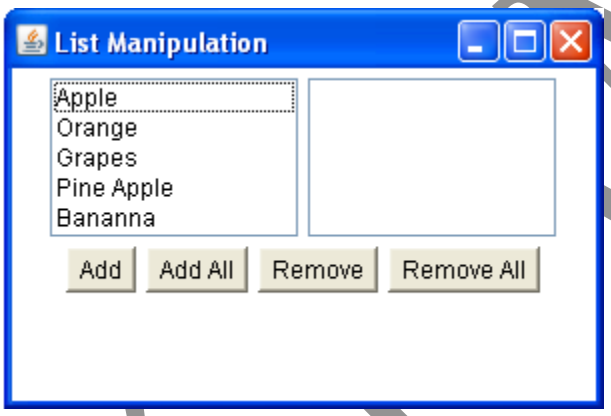
Exercise 9 : Develop the following application




```
import java.awt.*;
import java.awt.event.*;
class RadioDemo extends Frame implements ItemListener {
        Checkbox c1, c2, c3, c4;
        CheckboxGroup cbg;
        Panel p1;
        String msg;
        public RadioDemo(String title) {
                super(title);
                msg = "";
                cbg = new CheckboxGroup();
                p1 = new Panel();
                c1 = new Checkbox("Windows98", cbg, true);
                c2 = new Checkbox("WindowsNT", cbg, false);
                c3 = new Checkbox("Solaris", cbg, false);
                c4 = new Checkbox("macOS", cbg, false);
                c1.addItemListener(this);
                c2.addItemListener(this);
                c3.addItemListener(this);
                c4.addItemListener(this);
                p1.setLayout(new GridLayout(2, 2));
                p1.add(c1);
                p1.add(c2);
                p1.add(c3);
                p1.add(c4);
                add("South",p1);
                addWindowListener(new MyWindowAdapter());
```

```
                setSize(300, 300);
                show();
                msg = "Current Selection:"+cbg.getSelectedCheckbox().getLabel();
        }
        public static void main(String args[]) {
                RadioDemo r = new RadioDemo("Radios");
        }
        public void itemStateChanged(ItemEvent e) {
                msg = "Current selection:"+cbg.getSelectedCheckbox().getLabel();
                repaint();
        }
        public void paint(Graphics g){
                g.drawString(msg, 30, 60);
        }
}
class MyWindowAdapter extends WindowAdapter {
        public void windowClosing(WindowEvent we) {
                System.exit(0);
        }
}
```

Exercise 10: Develop the following application



Validations

1.On Clicking "Add", the selected item is removed from list1 and added to list2.

2.On clicking "Add All" all items of list1 are moved to list2. List1 should be cleared.

3.On clicking "Remove", the selected item from list2 is removed and added to list1.

4.On clicking "Remove All" all items of list2 are moved to list1. List2 should be cleared.

```
import java.awt.*;
import java.awt.event.*;
class Lists extends Frame implements ActionListener {
        List l1, l2;
        Button b1, b2, b3, b4;
        public Lists(String title) {
                super(title);
                setLayout(new FlowLayout());
                l1 = new List(5);
                l2 = new List(5);
                b1 = new Button("Add");
                b2 = new Button("Add All");
                b3 = new Button("Remove");
                b4 = new Button("Remove All");
                b1.addActionListener(this);
                b2.addActionListener(this);
                b3.addActionListener(this);
                b4.addActionListener(this);
                add(l1);
                add(l2);
                add(b1);
                add(b2);
                add(b3);
                add(b4);
                l1.addItem("Apple");
                l1.addItem("Orange");
                l1.addItem("Grapes");
                l1.addItem("Pine Apple");
                l1.addItem("Bananna");
                validate();
        }

        public void actionPerformed(ActionEvent e) {
                if (e.getSource() == b1) {
                        l2.addItem(l1.getSelectedItem());
                        l1.delItem(l1.getSelectedIndex());
                } else if (e.getSource() == b3) {
                        l1.addItem(l2.getSelectedItem());
                        l2.delItem(l2.getSelectedIndex());
                } else if (e.getSource() == b2) {
                        for (int i = 0; i < l1.getItemCount(); i++) {
                                l1.select(i);
                                l2.addItem(l1.getSelectedItem());
                        }
                        l1.clear();
                } else if (e.getSource() == b4) {
                        for (int i = 0; i < l2.getItemCount(); i++) {
                                l2.select(i);
                                l1.addItem(l2.getSelectedItem());
                        }
```

```
                l2.clear();
            }
        }

        public static void main(String args[]) {
            Lists s = new Lists("List Manipulation");
            s.setSize(300, 200);
            s.setVisible(true);
        }
}
```

Exercise11: Develop the following application

Validations

1.On clicking “Change Label”, the label should read Welcome to J2EE.and the buton,s caption should read”Undo”

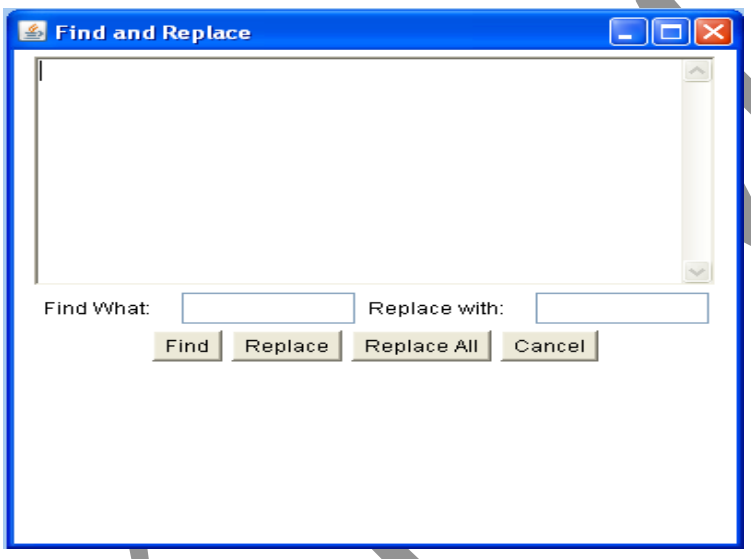
2.on clicking “Undo”the changes should be undone.

3.Onclicking “Exit”, the application is closed.

```
import java.awt.*;
import java.awt.event.*;
class ChangeLabelDemo extends Frame implements ActionListener {
        Label l1;
        Button b1, b2;
        Panel p1;
        public ChangeLabelDemo(String title) {
            p1 = new Panel();
            l1 = new Label("Welcome to java!", Label.CENTER);
            b1 = new Button("Change Label");
            b2 = new Button("Exit");
            b1.addActionListener(this);
            b2.addActionListener(this);
            p1.add(b1);
            p1.add(b2);
            add("North", l1);
            // add(11,BorderLayout.NORTH);
            add("South", p1);
        }
```

```
        public static void main(String args[]) {
                ChangeLabelDemo c = new ChangeLabelDemo("Change Label");
                c.setSize(200, 200);
                c.show();// c.set Visible(true);
        }
        public void actionPerformed(ActionEvent e) {
                if (e.getActionCommand().equals("Change Label")) {
                        l1.setText("Welcome toJ2EE!");
                        b1.setLabel("Undo");
                } else if (e.getActionCommand().equals("Undo")) {
                        l1.setText("Welcome to java!");
                        b1.setLabel("Change Label");
                } else if (e.getSource() == b2)
                        System.exit(0);
        }
}
```

Exercise 12: Develop the following application

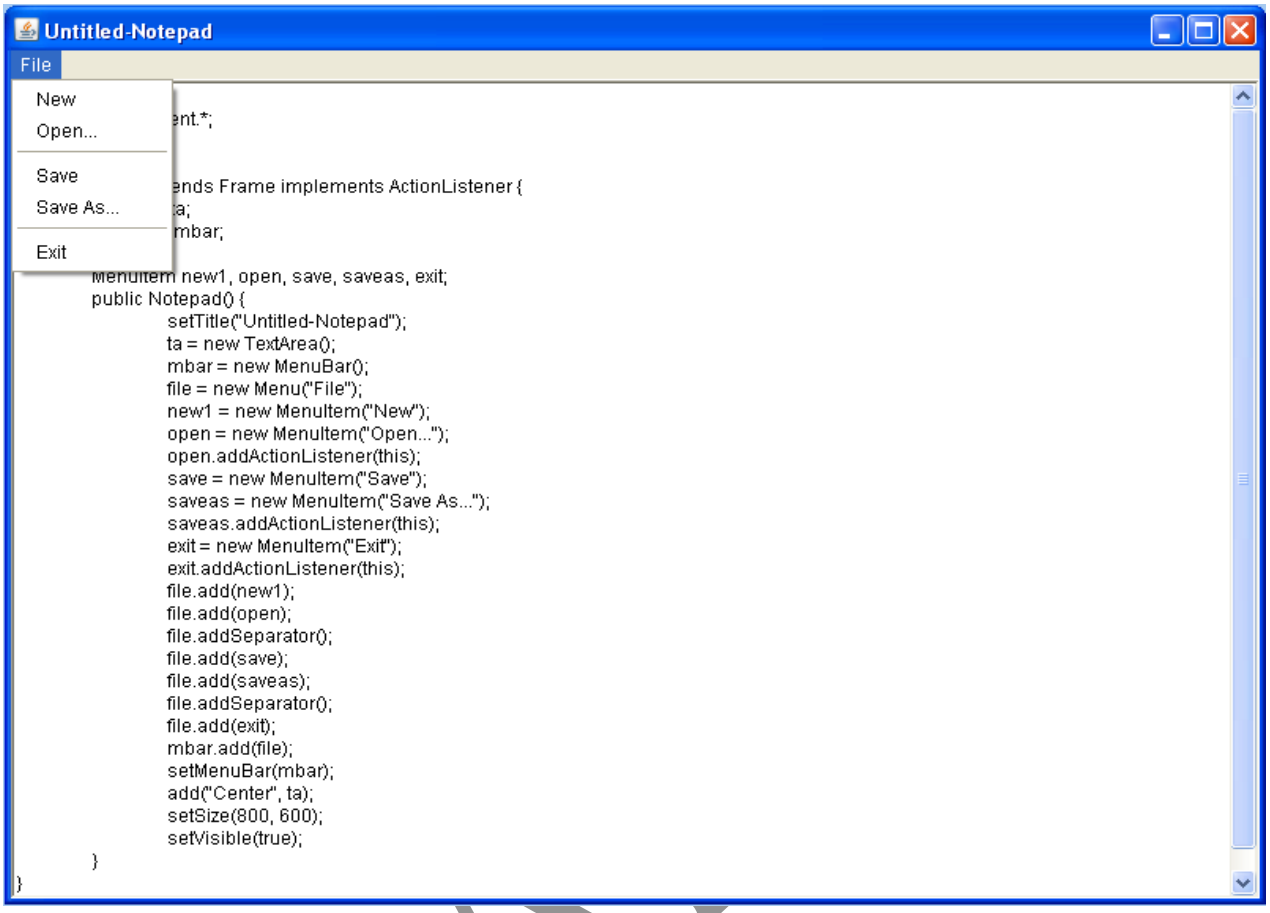


Note:
After typing in some text in the area.test the find and replace features of this application.

```
import java.awt.*;
import java.awt.event.*;
class Find extends Frame implements ActionListener {
        TextArea ta;
        TextField t1, t2;
        Label l1, l2;
        Button b1, b2, b3, b4;
        int startindex, len, i;
        public Find(String title) {
                super(title);
```

```
                i = startindex = len = 0;
                ta = new TextArea(10, 50);
                t1 = new TextField(10);
                t2 = new TextField(10);
                l1 = new Label("Find What:");
                l2 = new Label("Replace with:");
                b1 = new Button("Find");
                b2 = new Button("Replace");
                b3 = new Button("Replace All");
                b4 = new Button("Cancel");
                setLayout(new FlowLayout());
                b1.addActionListener(this);
                b2.addActionListener(this);
                b3.addActionListener(this);
                b4.addActionListener(this);
                add(ta);
                add(l1);
                add(t1);
                add(l2);
                add(t2);
                add(b1);
                add(b2);
                add(b3);
                add(b4);
        }
        public void actionPerformed(ActionEvent e) {
                String findwhat = new String("");
                String replacewith = new String("");
                String text = ta.getText();
                findwhat = t1.getText();
                len = findwhat.length();
                replacewith = t2.getText();
                if (e.getSource() == b1) {
                        startindex = text.indexOf(findwhat, i);
                        ta.select(startindex, startindex + len);
                        i = startindex + len;
                } else if (e.getSource() == b2)
                        ta.replaceRange(replacewith, startindex, startindex + len);
                else if (e.getSource() == b3)
                        for (int i = 0; i < text.length(); i = startindex + len) {
                                startindex = text.indexOf(findwhat, i);
                                ta.replaceRange(replacewith, startindex, startindex + len);
                        }
        }
        public static void main(String args[]) {
                Find f = new Find("Find and Replace");
                f.setSize(400, 400);
                f.setVisible(true);
        }
}
```

Exercise13:Developing an application that resembles Notepad Of Windows

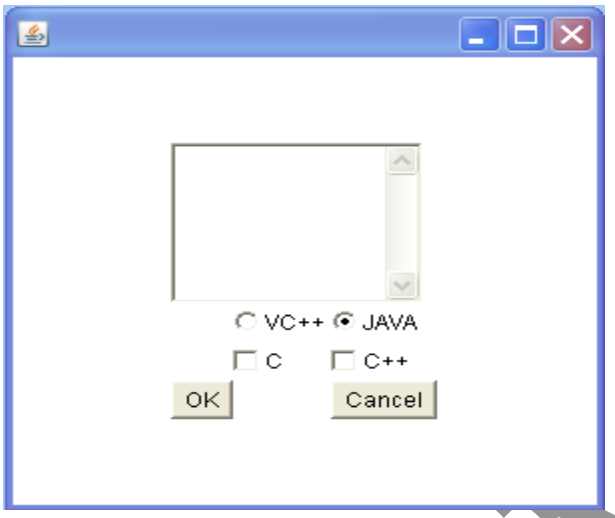



```
import java.awt.*;
import java.awt.event.*;
import java.io.*;
class Notepad extends Frame implements ActionListener {
        TextArea ta;
        MenuBar mbar;
        Menu file;
        MenuItem new1, open, save, saveas, exit;
        public Notepad() {
                setTitle("Untitled-Notepad");
                ta = new TextArea();
                mbar = new MenuBar();
                file = new Menu("File");
                new1 = new MenuItem("New");
                open = new MenuItem("Open...");
                open.addActionListener(this);
                save = new MenuItem("Save");
                saveas = new MenuItem("Save As...");
                saveas.addActionListener(this);
                exit = new MenuItem("Exit");
                exit.addActionListener(this);
```

```
            file.add(new1);
            file.add(open);
            file.addSeparator();
            file.add(save);
            file.add(saveas);
            file.addSeparator();
            file.add(exit);
            mbar.add(file);
            setMenuBar(mbar);
            add("Center", ta);
            setSize(800, 600);
            setVisible(true);
}
public static void main(String args[]) {
            new Notepad();
}
public void actionPerformed(ActionEvent e) {
            FileDialog fd;
            String file, dir, content;
            if (e.getSource() == open) {
                        fd = new FileDialog(this, "Open", FileDialog.LOAD);
                        fd.setVisible(true);
                        dir = fd.getFile();
                        file = fd.getFile();
                        try {
                                    FileInputStream fis = new FileInputStream(dir + file);
                                    content = "";
                                    int ch;
                                    while ((ch = fis.read()) != -1) {
                                                content = content + (char) ch;
                                    }
                                    ta.setText(content);
                                    setTitle(file + "-Notepad");
                                    fis.close();
                        } catch (Exception eee) {
                        }
            } else if (e.getSource() == saveas) {
                        fd = new FileDialog(this, "Save As", FileDialog.SAVE);
                        fd.setVisible(true);
                        dir = fd.getDirectory();
                        file = fd.getFile();
                        try {
                                    FileOutputStream fos = new FileOutputStream(dir + file);
                                    content = ta.getText();
                                    byte b[] = content.getBytes();
                                    fos.write(b);
                                    setTitle(file + "-Notepad");
                                    fos.close();
                        } catch (Exception eeee) {
                        }
```

```
                } else if (e.getSource() == exit) {
                        System.exit(0);
                }
        }
}
```

Exercise 14: Design the following user interface using GridBagLayout

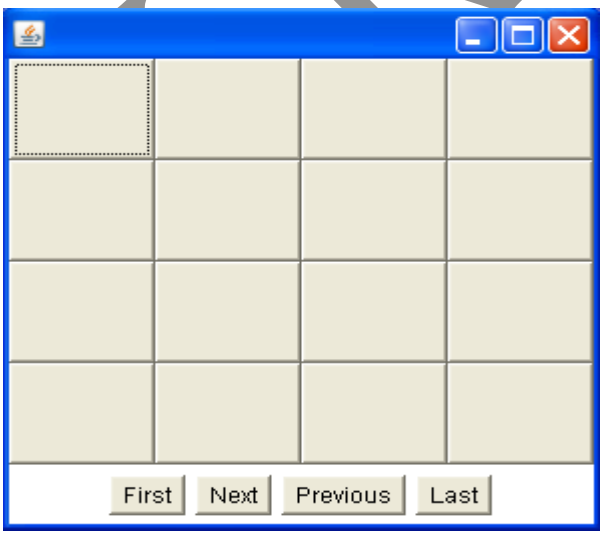



```
import java.awt.*;
class GridBagLayoutDemo extends Frame {
        Button b1, b2;
        TextField t1;
        Checkbox c1, c2, c3, c4;
        CheckboxGroup cbg;
        TextArea ta;
        GridBagLayout gb;
        GridBagConstraints gbc;
        public GridBagLayoutDemo() {
                gbc = new GridBagConstraints();
                cbg = new CheckboxGroup();
                gb = new GridBagLayout();
                b1 = new Button("OK");
                b2 = new Button("Cancel");
                ta = new TextArea(5, 15);
                t1 = new TextField();
                c1 = new Checkbox("C");
                c2 = new Checkbox("C++");
                c3 = new Checkbox("VC++", cbg, false);
                c4 = new Checkbox("JAVA", cbg, true);
                setLayout(gb);
                gbc.fill = GridBagConstraints.BOTH;
                addComponent(ta, 0, 0, 3, 1);
```

```
                gbc.fill = GridBagConstraints.HORIZONTAL;
                addComponent(b1, 3, 0, 1, 3);
                gbc.fill = GridBagConstraints.HORIZONTAL;
                addComponent(b2, 3, 2, 2, 3);
                gbc.fill = GridBagConstraints.HORIZONTAL;
                addComponent(c3, 1, 1, 1, 1);
                gbc.fill = GridBagConstraints.HORIZONTAL;
                addComponent(c4, 1, 2, 1, 1);
                gbc.fill = GridBagConstraints.HORIZONTAL;
                addComponent(c1, 2, 1, 1, 1);
                gbc.fill = GridBagConstraints.HORIZONTAL;
                addComponent(c2, 2, 2, 1, 1);
                // gbc.fill=GridBagConstraints.HORIZONTAL;
                // addComponent(ta,0,0,1,3);
                setSize(300, 300);
                setVisible(true);
        }
        public void addComponent(Component c, int row, int col, int nrow, int ncol) {
                gbc.gridx = col;
                gbc.gridy = row;
                gbc.gridwidth = ncol;
                gbc.gridwidth = nrow;
                gb.setConstraints(c, gbc);
                add(c);
        }
        public static void main(String args[]) {
                new GridBagLayoutDemo();
        }
}
```

Exercise15:Demonstring Card Layout

Note:
On clicking navigational buttons, you can move from one card to other card.



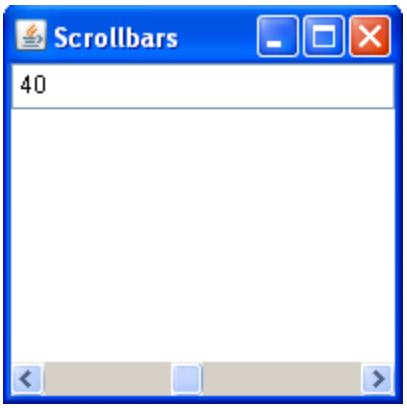
```
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
class Panel1 extends Panel {
	Button b[];
	public Panel1() {
		setLayout(new GridLayout(4, 4));
		b = new Button[16];
		for (int i = 0; i <= 15; i++) {
			b[i] = new Button("");
			add(b[i]);
		}
	}
}
class Panel2 extends Panel {
	Label l1, l2, l3;
	Button b1, b2;
	TextField t1, t2, t3;
	public Panel2() {
		setLayout(new FlowLayout());
		l1 = new Label("Num1:");
		l2 = new Label("Num2:");
		l3 = new Label("Res:");
		t1 = new TextField(10);
		t2 = new TextField(10);
		t3 = new TextField(10);
		b1 = new Button("Add");
		b2 = new Button("sub");
		add(l1);
		add(t1);
		add(l2);
		add(t2);
		add(l3);
		add(b1);
		add(b2);
	}
}
class Panel3 extends Panel {
	Label l1, l2;
	Button b1, b2;
	TextField t1, t2;
	public Panel3() {
		setLayout(new FlowLayout());
		l1 = new Label("User Id:");
		l2 = new Label("Password:");
		t1 = new TextField(10);
		t2 = new TextField(10);
		b1 = new Button("OK");
		b2 = new Button("Cancel");
		add(l1);
```

```
                add(t1);
                add(l2);
                add(t2);
                add(b1);
                add(b2);
        }
}
class CardLayoutDemo extends Frame implements ActionListener {
        Button b1, b2, b3, b4;
        Panel p = new Panel();
        CardLayout c = new CardLayout();
        Panel cards;// main panel
        public CardLayoutDemo() {
                cards = new Panel();
                cards.setLayout(c);
                b1 = new Button("First");
                b2 = new Button("Next");
                b3 = new Button("Previous");
                b4 = new Button("Last");
                p.add(b1);
                p.add(b2);
                p.add(b3);
                p.add(b4);
                Panel1 p1 = new Panel1();
                Panel2 p2 = new Panel2();
                Panel3 p3 = new Panel3();
                cards.add(p1, "panel1");
                cards.add(p2, "panel2");
                cards.add(p3, "panel3");
                add("Center", cards);
                add("South", p);
                b1.addActionListener(this);
                b2.addActionListener(this);
                b3.addActionListener(this);
                b4.addActionListener(this);
                setSize(300, 300);
                setVisible(true);
        }
        public void actionPerformed(ActionEvent e) {
                if (e.getSource() == b1)
                        c.first(cards);
                else if (e.getSource() == b2)
                        c.next(cards);
                else if (e.getSource() == b3)
                        c.previous(cards);
                else if (e.getSource() == b4)
                        c.last(cards);
        }
        public static void main(String args[]) {
                new CardLayoutDemo();
```

```
        }
}
```

Exercise 16: Develop the following application

It demonstrates the usage of scrollbar.



Validation:

On clicking scrollbar,its values shold appear in the text field.

```
import java.awt.*;
import java.awt.event.*;
class ScrollbarDemo extends Frame implements AdjustmentListener {
      TextField t1;
      Scrollbar s;
      public ScrollbarDemo() {
            addWindowListener(new MyWindowListener());
            setTitle("Scrollbars");
            t1 = new TextField(8);
            s = new Scrollbar(Scrollbar.HORIZONTAL);
            s.addAdjustmentListener(this);
            add("North", t1);
            add("South", s);
            setSize(200, 200);
            setVisible(true);
      }
      public void adjustmentValueChanged(AdjustmentEvent e) {
            t1.setText(String.valueOf(s.getValue()));
      }
      public static void main(String args[]) {
            new ScrollbarDemo();
      }
}
```

Exercise 17 :Creates a simple window.The window can be closed.

Demonstrates window Adapter.

```
import java.awt.*;
import java.awt.event.*;
```

```
class WindowAdapterDemo extends Frame {
	public WindowAdapterDemo() {
		addWindowListener(new MyWindowListener());
		setTitle("Window Adapter");
		setSize(300, 300);
		setVisible(true);
	}
	public static void main(String args[]) {
		new WindowAdapterDemo();
	}
}
class MyWiowListener extends WindowAdapter {
	public void windowClosing(WindowEvent e) {
		System.exit(0);
	}
}
```

Exercise 18:demonstrates mouse events

```
import java.awt.*;
import java.awt.event.*;
class MouseEvents extends Frame implements MouseListener, MouseMotionListener {
	String msg = "Move the mouse now";
	public MouseEvents() {
		addMouseListener(this);
		addMouseMotionListener(this);
		setTitle("Mouse Events");
		setBackground(Color.cyan);
		setVisible(true);
	}
	public void mousePressed(MouseEvent e) {
		msg = "Mouse pressed at" + e.getX() + "," + e.getY();
		repaint();
	}
	public void mouseReleased(MouseEvent e) {
		msg = "Mouse released at" + e.getX() + "," + e.getY();
		repaint();
	}
	public void mouseClicked(MouseEvent e) {
		msg = "Mouse clicked at" + e.getX() + "," + e.getY();
		repaint();
	}
	public void mouseEntered(MouseEvent e) {
		msg = "Mouse entered at" + e.getX() + "," + e.getY();
		repaint();
	}
	public void mouseExited(MouseEvent e) {
		msg = "Mouse exited at" + e.getX() + "," + e.getY();
		repaint();
	}
	public void mouseMoved(MouseEvent e) {
		msg = "Mouse moved at" + e.getX() + "," + e.getY();
		repaint();
	}
	public void mouseDragged(MouseEvent e) {
```

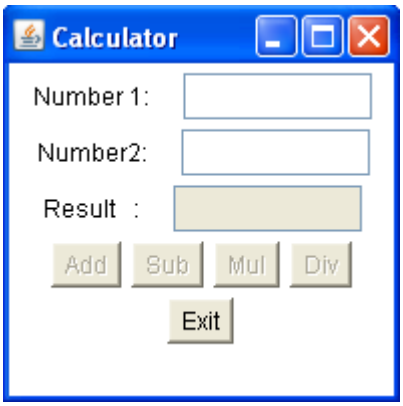
```
                msg = "Mouse dragged at" + e.getX() + "," + e.getY();
                repaint();
        }
        public void paint(Graphics g) {
                Font f = new Font("Times New Roman", Font.BOLD, 40);
                g.setFont(f);
                g.setColor(Color.blue);
                g.drawString(msg, 50, 50);
        }
        public static void main(String args[]) {
                new MouseEvents();
        }
}
```

Exercise 19 :Create a window that captureskey strokes

```
import java.awt.*;
import java.awt.event.*;
class KeyEvents extends Frame implements KeyListener {
        String msg = "";
        public KeyEvents() {
                addKeyListener(this);
                setTitle("Key Events");
                setSize(300, 300);
                setBackground(Color.gray);
                setVisible(true);
        }
        public void keyTyped(KeyEvent e) {
                msg += e.getKeyChar();
                repaint();
        }
        public void keyPressed(KeyEvent e) {
        }
        public void keyReleased(KeyEvent e) {
        }
        public void paint(Graphics g) {
                Font f = new Font("times New Roman", Font.BOLD, 40);
                g.setFont(f);
                g.setColor(Color.blue);
                g.drawString(msg, 50, 50);
        }
        public static void main(String args[]) {
                new KeyEvents();
        }
}
```

Exercise 20: Demonstrates FocusListener and TextListener

Develop the following applications with given validations.



Validations:

1.Disable all buttons except "Exit".

2.Buttons should be enabled when only the first two text fields ae not empty.

3.Provide appropriate action for all buttons.

First two text fields should not be left blank.

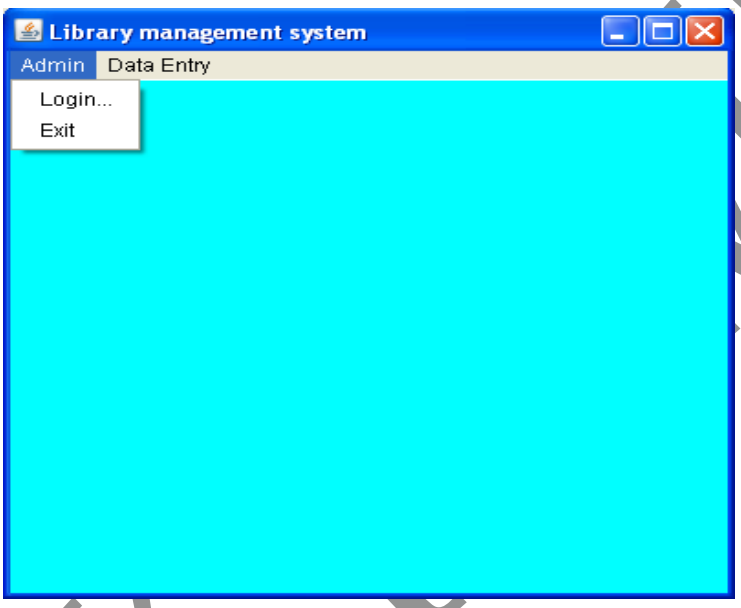
```
import java.awt.*;
import java.awt.event.*;
class Calculator extends Frame implements ActionListener, TextListener,
                        FocusListener {
        Label l1, l2, l3;
        TextField t1, t2, t3;
        Button b1, b2, b3, b4, b5;
        int x, y, res = 0;
        public Calculator(String title) {
                super(title);
                setLayout(new FlowLayout());
                l1 = new Label("Number 1:");
                l2 = new Label("Number2:");
                l3 = new Label("Result   :");
                t1 = new TextField(10);
                t1.addFocusListener(this);
                t2 = new TextField(10);
                t3 = new TextField(10);
                t3.setEditable(false);
                b1 = new Button("Add");
                b2 = new Button("Sub");
                b3 = new Button("Mul");
                b4 = new Button("Div");
                b5 = new Button("Exit");
                b1.setEnabled(false);
                b2.setEnabled(false);
                b3.setEnabled(false);
                b4.setEnabled(false);
                b1.addActionListener(this);
```

```
        b2.addActionListener(this);
        b3.addActionListener(this);
        b4.addActionListener(this);
        b5.addActionListener(this);
        t1.addTextListener(this);
        t2.addTextListener(this);
        add(l1);
        add(t1);
        add(l2);
        add(t2);
        add(l3);
        add(t3);
        add(b1);
        add(b2);
        add(b3);
        add(b4);
        add(b5);
}
public static void main(String args[]) {
        Calculator c = new Calculator("Calculator");
        c.setSize(200, 200);
        c.setVisible(true);
}
public void actionPerformed(ActionEvent e) {
        x = Integer.parseInt(t1.getText());
        y = Integer.parseInt(t2.getText());
        if (e.getSource() == b1)
                res = x + y;
        else if (e.getSource() == b2)
                res = x - y;
        else if (e.getSource() == b3)
                res = x * y;
        else if (e.getSource() == b4)
                res = x / y;
        else if (e.getSource() == b5)
                System.exit(0);
        t3.setText(String.valueOf(res));
}
public void copy() {
        x = Integer.parseInt(t1.getText());
        y = Integer.parseInt(t2.getText());
}
public void textValueChanged(TextEvent te) {
        if (!t1.getText().equals("") & (!t2.getText().equals(""))) {
                b1.setEnabled(true);
                b2.setEnabled(true);
                b3.setEnabled(true);
                b4.setEnabled(true);
        } else {
                b1.setEnabled(false);
```

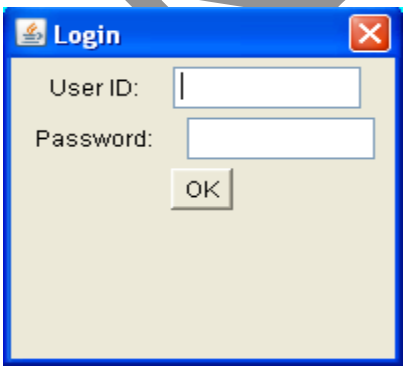
```
                    b2.setEnabled(false);
                    b3.setEnabled(false);
                    b4.setEnabled(false);
                }
        }
        public void focusGained(FocusEvent fe) {
        }
        public void focusLost(FocusEvent fe) {
                if (t1.getText().equals("")) {
                    javax.swing.JOptionPane.showMessageDialog(this,
                                "The field should not be left blank");
                }
        }
}
```

Exercise 21: Demonstrates dialog boxes

Develop the following main application.(Library.java)



On clicking Login…,The following login dialog should be displayed to authenticate the user.(Login.java)



Library.java

```
import java.awt.*;
import java.awt.event.*;
class Library extends Frame implements ActionListener {
	MenuBar mbar;
	Menu admin, dataEntry;
	static MenuItem login, exit, book, member, issue;
	public Library() {
		setTitle("Library management system");
		admin = new Menu("Admin");
		dataEntry = new Menu("Data Entry");
		login = new MenuItem("Login...");
		exit = new MenuItem("Exit");
		book = new MenuItem("books...");
		member = new MenuItem("Members...");
		issue = new MenuItem("Issues...");
		login.addActionListener(this);
		exit.addActionListener(this);
		admin.add(login);
		admin.add(exit);
		dataEntry.add(book);
		dataEntry.add(member);
		dataEntry.add(issue);
		mbar = new MenuBar();
		mbar.add(admin);
		mbar.add(dataEntry);
		disable();
		setBackground(Color.cyan);
		setMenuBar(mbar);
		setSize(400, 400);
		setVisible(true);
	}
	public void actionPerformed(ActionEvent e) {
		if (e.getSource() == login) {
			Login x = new Login(new Frame(), "Login", true);
		} else if (e.getSource() == exit)
			System.exit(0);
	}
	public static void main(String args[]) {
		new Library();
	}
	public void disable() {
		book.setEnabled(false);
		member.setEnabled(false);
		issue.setEnabled(false);
	}
	public static void enableItems() {
		book.setEnabled(true);
```

```
                member.setEnabled(true);
                issue.setEnabled(true);
        }
}
```

Login.java:

```
import java.awt.*;
import java.awt.event.*;
class Login extends Dialog implements ActionListener {
        // declaring controls
        Label l1, l2;
        TextField t1, t2;
        Button b1, b2;
        // constructor
        public Login(Frame f, String title, Boolean b) {
                super(f, title, b);
                // creating controls
                l1 = new Label("User ID:");
                l2 = new Label("Password:");
                t1 = new TextField(10);
                t2 = new TextField(10);
                t2.setEchoChar('*');
                b1 = new Button("OK");
                b2 = new Button("Cancel");
                // registering controls with listener
                b1.addActionListener(this);
                b2.addActionListener(this);
                // setting flow layout
                setLayout(new FlowLayout());
                // adding controls to frame
                add(l1);
                add(t1);
                add(l2);
                add(t2);
                add(b1);
                setSize(200, 200);
                setVisible(true);
        }
        public void actionPerformed(ActionEvent e) {
                if (e.getSource() == b1) {
                        if (t1.getText().equals("active") && t2.getText().equals("Aspire"))
                                javax.swing.JOptionPane.showMessageDialog(this, "Valid User");
                        else
                                javax.swing.JOptionPane.showMessageDialog(this, "Invalid User");
```

```
                } else
                        dispose();
        }// actionPerformed()
}// class
```

9.APPLETS

Exercise 1:Login Applet which authenticates the user

```
/*< applet code= Login Applet height=200width=200></applet> */
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class LoginApplet extends Applet implements ActionListener {
        Label l1, l2;
        TextField t1, t2;
        Button b1, b2;
        public void init() {
                l1 = new Label("User ID:");
                l2 = new Label("passwors:");
                t1 = new TextField(10);
                t2 = new TextField(10);
                t2.setEchoChar('*');
                b1 = new Button("Submit");
                b2 = new Button("Reset");
                b1.addActionListener(this);
                b2.addActionListener(this);
                add(l1);
                add(t1);
                add(l2);
                add(t2);
                add(b1);
                add(b2);
                setBackground(Color.cyan);
        }
        public void actionPerformed(ActionEvent e) {
                if (e.getSource() == b1) {
                        if (t1.getText().equals("aspire")
                                        && t2.getText().equals("technologies"))
                                showStatus("valid User");
                        else
                                showStatus("Invalid User");
                } else {
                        t1.setText("");
                        t2.setText("");

                }
        }
}
```

Testing applet

Appletviewer LoginApplet.java

Embedding applet into HTML page

```
<HTML>
<BODY bgcolor=cyan>
<center>
<applet  code= Login Applet  width=200 height=200></applet>
</body>
</html>
```

Open the above file in browser to test the applet.

Exercise 2: drawing images on Applet

```
import java.awt.*;
import java.applet.*;
/*<applet code=image Demo height=300 width=300></applet>*/
public class ImageDemo extends Applet {
        Image i, j, k;
        public void init() {
                i = getImage(getCodeBase(), "javalogo.gif");
                j = getImage(getCodeBase(), "globe.gif");
                k = getImage(getCodeBase(), "thumbsup.gif");
        }
        public void paint(Graphics g) {
                g.drawImage(i, 20, 20, this);
                g.drawImage(j, 60, 60, this);
                g.drawImage(k, 150, 100, this);
        }
}
```

Note:
Before running the above applet ensure that images are available in the folder where you save your applet.

Exericse 3: Passing runtime parameters to applet:

```
import java.awt.*;
import java.applet.*;
/*<applet code=image Demo param height=300></applet>*/
public class ImageDemoParam extends Applet {
        Image i;
        public void init() {
                String imagename = getParameter("Image");
                i = getImage(getCodeBase(), imagename);
        }
        public void paint(Graphics g) {
                g.drawImage(i, 20, 20, this);
        }
}
```

HTML Page

```
<HTML>
<HEAD>
        <TITLE>Applet paramenters</TITLE>
<HEAD>
<BODY  BGCOLOR=orange text=blue>
<FONT COLOR=RED SIZE=10>
<MARQUEE>
</FONT>
<center>
<APPLET CODE=Applet Para height=200 width=200>
<param name=”image” value=”duke.gif”>
</applet>
<center>
</BODY></HTML>
```

On opening this HTML file in browser, the applet shows duke.gif.Try to modify the source code(View->Source) to change the value for image parmeter and observe the difference.

Exercise 4: Displays various graphics on the applet:

```
import java.applet.*;
import java.awt.*;
/*<applet code=paint Applet width=300 height=300></applet*/
public class PaintApplet extends Applet {
        public void init() {
                setBackground(Color.yellow);
                setForeground(Color.blue);
        }
        public void paint(Graphics g) {
                g.fillRect(50, 50, 50, 50);
                g.setColor(Color.red);
                g.fillOval(110, 110, 60, 60);
                Font f = new Font("Times new Roman", Font.BOLD, 40);
                g.setColor(Color.green);
                g.drawString("Happiness is an attitude", 150, 200);
                g.drawLine(100, 100, 300, 300);
        }
}
```

Exercise 5: Conneeting to Intranet/Internet websites:

```
/*<applet code=Url Test height=300width=300></applet*/
import java.awt.event.*;
import java.awt.*;
import java.applet.*;
import java.net.*;
public class UrlTest extends Applet implements ActionListener {
```

```
        URL url;
        Button b;
        public void init() {
                setBackground(Color.yellow);
                b = new Button("yahoo");
                b.addActionListener(this);
                add("Center", b);
                try {
                        url = new URL("http://www.yaoo.com");
                } catch (MalformedURLException e) {
                }

        }
        public void actionPerformed(ActionEvent ae) {
                if (ae.getSource() == b)
                        getAppletContext().showDocument(url, "");
        }
}
```

Develop an HTML file to embed this applet.

On clicking yahoo button,you will get connected to yahoo.com. In absence of Internet connection,you can try to connect to local web server by changing the URL as follows.

URL url=new URL(http://localhost:8080/sitename/pagename.html);

Note:

Take the help of lab –coordinator to find name of web application that is currently running in local or intranet server.

Exercise 6: digital Clock Applet

```
import java.awt.Graphics;
import java.awt.Font;
import java.awt.Color;
import java.util.Date;
/* <applet code=*DigitalClock .alss"width=400height=400> </applet> */
public class DigitalClock extends java.applet.Applet implements Runnable {
        Font theFont = new Font("TimesRoman", Font.BOLD, 34);
        Date theDate;
        Thread runner;
        public  void  start() {
                if (runner == null) {
                        runner = new Thread(this);
                        runner.start();
                }
        }
        public  void  stop() {
```

```
            if (runner != null) {
                runner.stop();
                runner = null;
            }
        }
        public  void  run() {
            while (true) {
                theDate = new  Date();
                repaint();
                try {
                    Thread.sleep(1000);
                } catch (InterruptedException e) {
                }
            }
        }
        public  void  paint(Graphics g) {
            g.setFont(theFont);
            g.setColor(Color.red);
            g.draw String(theDate.toString(), 10, 50);
        }
}
```

Exercise 7: Bouncing Ball game applet

```
/*<applet code=Bouncing Ball height=300width=300></applet>*/
import java.awt.*;
import java.applet.*;
public class BouncingBall extends Applet implements Runnable {
        int top, left, w;
        Thread t;
        int i;
        public void init() {
            w = 30;
            t = new Thread(this);
            t.start();
        }
        public void paint(Graphics g) {
            while (true) {
                i = (int) (Math.random() * 1000) / 15;
                if (i != 5) {
                    top = (int) (Math.random() * 1000) / (5 - i);
                    left = (int) (Math.random() * 1000) / (5 - i);
                }
                if (i >= 0 && i <= 6)
                    break;
            }
            g.setColor(Color.red);
            g.fillOval(top, left, w, w);
```

```
	}
	public void run() {
		while (true) {
			try {
				repaint();
				Thread.sleep(500);
			} catch (InterruptedException e) {
			}
		}
	}
}
```

Excerise 8: Ping Pong game applet

```
/*<applet code=ping pongheight=300width=></applet*/
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class PingPong extends Applet implements Runnable, ActionListener {
	int top, left, w;
	Thread t;
	int i;
	Boolean threadstop = false;
	int score = 0;
	Button b1 = new Button("Start");
	Button b2 = new Button("End");
	int x1, x2;
	Panel p = new Panel();
	public void init() {
		setLayout(new BorderLayout());
		w = 30;
		p.add(b1);
		p.add(b2);
		add("South", p);
		b1.addActionListener(this);
		b2.addActionListener(this);
	}
	public void actionPerformed(ActionEvent e) {
		if (e.getSource().equals(b1)) {
			t = new Thread(this);
			t.start();
		} else if (e.getSource().equals(b2)) {
			threadstop = true;
			removeAll();
			repaint();
		}
	}
	public boolean mouseDown(Event e, int a1, int a2) {
		x1 = a1;
		x2 = a2;
		repaint();
```

```
                return true;
        }
        public boolean mouseUp(Event e, int a3, int a4) {
                x1 = a3;
                x2 = a4;
                if ((top >= x1) && (left <= x1 + 150))
                        score = score + 10;
                repaint();
                return true;
        }
        public boolean mouseDrag(Event e, int a3, int a4) {
                x1 = a3;
                x2 = a4;
                if ((top >= x1) && (left <= x1 + 150))
                        score = score + 10;
                repaint();
                return true;
        }
        public void paint(Graphics g) {
                if (threadstop == false) {
                        while (true) {
                                i = (int) (Math.random() * 1000) / 15;
                                if (i != 5) {
                                        top = (int) (Math.random() * 1000) / (5 - i);
                                        left = (int) (Math.random() * 1000) / (5 - i);
                                }
                                if (i >= 0 && i <= 6)
                                        break;
                        }
                        g.setColor(Color.red);
                        g.fillOval(top, left, w, w);
                        g.setColor(Color.black);
                        g.fillRect(x1, x2, 150, 30);
                } else if (threadstop == true) {
                        setBackground(Color.cyan);
                        g.setFont(new Font("Helvetic", Font.BOLD, 20));
                        g.drawString("Game Over!", 20, 200);
                        g.drawString("Score=" + score, 20, 220);
                }
        }
        public void run() {
                if (threadstop == false) {
                        try {
                                repaint();
                                Thread.sleep(750);
                        } catch (InterruptedException e) {
                        }
                }
        }
}
```

Exercise 9: Scrolling Banner applet

```
/*applet code=VerticslScroll height=300 width=300></applet>*/
import java.awt.*;
import java.applet.*;
public class VerticalScroll extends Applet implements Runnable {
        String message = "Happinss is an attitude....";
        int x, y;
        Dimension d;
        public void init() {
                x = 10;
                y = 30;// set inital coordinates for message
                d = getSize();// get the applet size
                Thread t = new Thread(this);// create new thread
                t.start();
        }
        public void run() {
                while (true) {
                        try {
                                repaint();
                                if (y >= d.height)
                                        // if end of screen isreached go back to start
                                        y = 30;
                                else
                                        y++;
                                Thread.sleep(50);
                        } catch (InterruptedException e) {
                        }
                }
        }
        public void paint(Graphics g) {
                g.setFont(new Font("Helvetica", Font.BOLD | Font.ITALIC, 24));
                g.setColor(Color.magenta);
                g.drawString(message, x, y);
        }
}
```

Exercise 10: simple animation applet

```
/*<applet code=TwoImages height=300 width=300>*/
import java.awt.*;
import java.applet.*;
public class TwoImages extends Applet implements Runnable {
        Image i1, i2;
        int x1, y1, x2, y2;
        public void init() {
                Dimension d = getSize();
                x1 = 5;
                y1 = 3;
```

```
                x2 = d.width;
                y2 = 3;
                i1 = getImage(getCodeBase(), "duke.gif");
                i2 = getImage(getCodeBase(), "duke.gif");
                Thread t = new Thread(this);
                t.start();
        }
        public void run() {
                while (true) {
                        x1 = x1 + 3;
                        y1 = y1 + 3;
                        x2 = x2 - 3;
                        y2 = y2 + 3;
                        try {
                                Thread.sleep(100);
                        } catch (InterruptedException e) {
                                break;
                        }
                        repaint();
                }
        }
        public void paint(Graphics g) {
                g.drawImage(i1, x1, y1, this);
                g.drawImage(i2, x2, y2, this);
        }
}
```

Try It Out

Rewrite the Puzzle game that you have come across in AWT using Applet as container. Develop an HTML page to embedand test the game applet.

==========