SRM MADURAI

COLLEGE FOR ENGINEERING AND TECHNOLOGY

Approved by AICTE, New Delhi | Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LABORATORY MANUAL

Sub.Code : CS3381

Sub.Name :Object Oriented Programming Laboratory

Regulation : R2021

Prepared By,

Dr.S.J.Subashini

Associate Professor/CSE

Verified By,

Dr.C.Callins Christiyana

Prof & HoD/CSE

INSTITUTE VISION
To become a centre of excellence in preparing engineering with excellent technical, scientific research and entrepreneurial abilities to contribute to the society.

INSTITUTE MISSION	
1	Providing comprehensive learning environment
2	Imparting state-of-the-art technology to fulfil the needs of the students and Industry
3	Establishing Industry-Institute alliance for bilateral benefits
4	Promoting Research and Development activities
5	Offering student lead activities to inculcate ethics, social responsibilities, entrepreneurial, and leadership skills

DEPARTMENT VISION
To become a centre of excellence in technical education and scientific research in the field of Computer Science and Engineering for the wellbeing of the society.

DEPARTMENT MISSION	
1	Producing graduates with a strong theoretical and practical in computer technology to meet the Industry expectation.
2	Offering holistic learning ambience for faculty and students to investigate, apply and transfer knowledge.
3	Inculcating interpersonal traits among the students leading to employability and entrepreneurship.
4	Establishing effective linkage with the Industries for the mutual benefits
5	Strengthening Research activities to solve the problems related to industry and society.

SYLLABUS

CS3381	OBJECT ORIENTED PROGRAMMING LABORATORY	L	T	P	C
		0	0	3	1.5

COURSE OBJECTIVES :

- To build software development skills using java programming for real-world applications.
- To understand and apply the concepts of classes, packages, interfaces, inheritance, exception handling and file processing.
- To develop applications using generic programming and event handling

EXPERIMENTS

1. Solve problems by using sequential search, binary search, and quadratic sorting algorithms (selection, insertion)
2. Develop stack and queue data structures using classes and objects.
3. Develop a java application with an Employee class with Emp_name, Emp_id, Address, Mail_id, Mobile_no as members. Inherit the classes, Programmer, Assistant Professor, Associate Professor and Professor from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 97% of BP as DA, 10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club funds. Generate pay slips for the employees with their gross and net salary.
4. Write a Java Program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method printArea( ) that prints the area of the given shape.
5. Solve the above problem using an interface.
6. Implement exception handling and creation of user defined exceptions.
7. Write a java program that implements a multi-threaded application that has three threads. First thread generates a random integer every 1 second and if the value is even, the second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of the cube of the number. 8. Write a program to perform file operations.
8. Develop applications to demonstrate the features of generics classes.
9. Develop applications using JavaFX controls, layouts and menus.
10. Develop a mini project for any application using Java concepts.

TOTAL: 45 Periods

CONTENT BEYOND SYLLABI: Creating a login form using servlet

COURSE OUTCOMES:

On completion of the course, students will be able to:

CO1: Design and develop java programs using object oriented programming concepts

CO2 : Develop simple applications using object oriented concepts such as package, exceptions CO3: Implement multithreading, and generics concepts

CO4 : Create GUIs and event driven programming applications for real world problems

CO5: Implement and deploy web applications using Java

SOFTWARE AND HARDWARE REQUIREMENT

Operating Systems: Linux / Windows

Front End Tools: Eclipse IDE / Netbeans IDE

LIST OF EXPERIMENTS

ExNo:1(a) **SEQUENTIAL SEARCH**

Date

AIM:

To develop a java program for Sequential search using array.

ALGORITHM:

1. We are searching the key in the array.
2. Read the array length and store the value into the variable len, read the elements using the Scanner class method and store the elements into the array array[].
3. 3. Read the key value and search for that key in the array.
4. Run the for loop for i = 0 to i < length of the array.
5. Compare array[i] with the key, If any one of the elements of an array is equal to the key then print the key and position of the key.

PROGRAM:

```
import java.util.Scanner;
class Linear {
   public static void main(String args[]) {
      int i, len, key, array[];
      Scanner input = new Scanner(System.in);
      System.out.println("Enter Array length:");
      len = input.nextInt();
      array = new int[len];
      System.out.println("Enter " + len + " elements");
      for (i = 0; i < len; i++) {
         array[i] = input.nextInt();
      }
      System.out.println("Enter the search key value:");
      key = input.nextInt();
      for (i = 0; i < len; i++) {
         if (array[i] == key) {
            System.out.println(key + " is present at location " + (i + 1));
            break;
         }
      }
      if (i == len)
         System.out.println(key + " doesn't exist in array.");
   }
}
```

OUTPUT:

```
Enter Array length: 5
Enter 5 elements
8
3
56
4
8
Enter the search key value: 1
1 doesn't exist in
array. Enter Array
length:
3
Enter 3 elements
11
21
31
Enter the search key value: 11
11 is present at location 1
```

ExNo: 1(b) **BINARY SEARCH**
Date:

AIM:

To develop a java program for Binary search using array.

ALGORITHM:

1. Start the program
2. Create an object of binary class
3. Create a sorted array
4. Get input from user for element to be searched
5. Call the binary search method pass arguments: array, element, index of first and last element
6. Create the Binary search function definition
7. Checking the condition **while** (low <= high) Repeat until the pointers low and high meet each other
8. Get index of mid element **int** mid = low + (high - low) / 2, if element to be searched is the mid element **return** mid.if element is less than mid element search only the left side of mid. if element is greater than mid element search only the right side of mid, else return -1.
9. Print the result and stop the process.

PROGRAM:

```
import java.util.Scanner;

public class Binary {
   int binarySearch(int array[], int element, int low, int high) {
      while (low <= high) {
         int mid = low + (high - low) / 2;
         if (array[mid] == element)
            return mid;
         if (array[mid] < element)
            low = mid + 1;
         else
            high = mid - 1;
      }
      return -1;
   }

public static void main(String args[]) {
   Binary obj = new Binary();
   int[] array = {3, 4, 5, 6, 7, 8, 9};
   int n = array.length;
   Scanner input = new Scanner(System.in);
   System.out.println("Enter element to be searched:");
   // element to be searched
   int element = input.nextInt();
   input.close();
```

```
        int result = obj.binarySearch(array, element, 0, n - 1);
        if (result == -1)
         System.out.println("Not found");
      else
         System.out.println("Element found at index " + result);
   }
}
```

OUTPUT:

```
Enter element to be
searched: 8
Element found at index 5
```

ExNo: 1(c) **SELECTION SORTING**
Date:

AIM:
To Develop Java program for Selecting Sorting Using Array.

ALGORITHM:

1. Start the process
2. Entered numbers will store in to the int array a[] using for loop with the structure for(i=0; i < n; i++).
3. Printarray(int a[]) will print the numbers, from the index i=0 to i<length of the array. **4.** Sort(int a[]) will sort the numbers in ascending order. The inner loop will find the next least number to the previous number and the outer loop will place the least number in proper position in the array.
Given numbers are 9, 0, 1, 23, 99, 5.
a) The inner loop will compare the first two numbers 9,0 , the least number is 0, then the loop compares 0 with 1, 23, 99, 5. There is no least number available than 0. So outer loop swap the 9,0 numbers. Then the series is 0, 9, 1, 23, 99, 5.
b) Now the inner loop compares the 9,1, the number 1 is the least than 9, then compare 1 with 23, 99, 5. Compare with the next elements, 1 is the least number. Outer loop swap the numbers 9,1. Now the series is 0, 1, 9, 23, 99, 5.
c) Compare 9,23, then the least number is 9, find the least number than 9. In this series 5 is least compare with 9, so the outer loop swap the numbers 9,5. The series is 0, 1, 5, 23, 99, 9. **d)**Compare 23,99, the least number is 23, find the least number than 23, 9 is the least number in the remaining series, Outer loop swap the numbers 23,9.Now the series is 0,1,5,9,99,23. **e)** Compare 99 with 23, 23 is the least number, swap the numbers 23,99. After selection sort, the number series is 0, 1, 5, 9, 23, 99.
5. Print the result.

PROGRAM:

```
import java.util.Scanner;
public class SSort {
   public static void Sort(int a[]) {
      int n = a.length, i, j, p, temp;
      for (i = 0; i < n - 1; i++) {
         p = i;
         for (j = i + 1; j < n; j++) {
            if (a[p] > a[j])
               p = j;
         }
              temp = a[p];
             a[p] = a[i];
             a[i] = temp;
          }
       }
```

```
    public static void printarray(int a[]) {
        for (int i = 0; i < a.length; i++) {
            System.out.print(a[i] + " ");
        }
    }

    public static void main(String[] args) {
        int n, res, i;
        Scanner s = new Scanner(System.in);
        System.out.print("Enter number of elements in the array:");
        n = s.nextInt();
        int a[] = new int[n];
        System.out.println("Enter " + n + " elements ");
        for (i = 0; i < n; i++) {
            a[i] = s.nextInt();
        }
        System.out.println("elements in array ");
        printarray(a);
        Sort(a);
        System.out.println("\nelements after sorting");
        printarray(a);
    }
}
```

OUTPUT:

Enter number of elements in the
array:6 Enter 6 elements
9

ExNo: 1(d) **INSERTION SORTING**
Date:

AIM:

To develop Java program for Insertion Sorting using Array.

ALGORITHM:

1. We are using an array for insertion sort.
2. The print method will print the array elements, the sort method will sort the array elements.
3. The elements in the array are 9, 5, 0, 1, 8.For 1st iteration, the inner loop compares the number with the previous number, if the previous number is greater than this number then shift the least number to left. For i=1, inner loop compares the numbers 5<9, then 5 will be shifted to left. Then the series is 5, 9, 0, 1, 8. In this sorted subarray is 5,9.for i=2, the inner loop compares the numbers 0<9, shift 0 to left, compare 0<5, shift 0 to left.
4. Then the sorted subarray is 0, 5, 9. The series is 0, 5, 9, 1, 8.For i=3, the inner loop will compare the numbers 1<9, shift 1 to left, compare 1<5, shift 1 to left, compare 1,0. The sorted subarray is 0, 1, 5, 9 and the series is 0, 1, 5, 9, 8.
For i=4, the inner loop will compare the numbers 8<9, shift 8 to left. The sorted series is 0, 1, 5, 8, 9.
5. Stop the Process.

PROGRAM:

```
import java.util.Scanner;

public class ISort {
   public static void Sort(int a[]) {
      int n = a.length, i, j, temp;
      for (i = 1; i < n; i++) {
         for (j = i - 1; j >= 0 && a[j + 1] < a[j]; j--) {
            temp = a[j + 1];
            a[j + 1] = a[j];
            a[j] = temp;
         }
      }
   }

   public static void printarray(int a[]) {
      for (int i = 0; i < a.length; i++) {
         System.out.print(a[i] + " ");
      }
   }

   public static void main(String[] args) {
      int n, i;
      Scanner s = new Scanner(System.in);
      System.out.print("Enter number of elements in the array:");
      n = s.nextInt();
      int a[] = new int[n];
      System.out.println("Enter " + n + " elements ");
```

```
        for (i = 0; i < n; i++) {
            a[i] = s.nextInt();
        }

        System.out.println("elements in array ");
        printarray(a);
        Sort(a);
        System.out.println("\nelements after sorting");
        printarray(a);
    }
}
```

OUTPUT:

```
Enter number of
elements in the array:5
Enter 5 elements
9
5
0
1
8
elements in array    9
5 0 1 8
elements after sorting 0 1 5
8 9
```

Sample Viva Questions:

1. What is Class and object?
2. What is an array?
3. How to declare an array in java?
4. What is Sequential Searching?
5. What is Method?

Augmented Questions:

1. Write a Java program to sum the values of an array.
2. Write a Java program to find a missing number in an array

ExNo: 2(a) **STACK IMPLENETATION USING CLASS AND OBJECT**
Date:

AIM:

To develop Java program for Stack Implementation using Class and Object.

ALGORITHM:

1. Create stack and Store elements in stack for push pop operation
2. Push the elements to the top of stack, before push element in stack should stack is not full
3. Pop the elements from the stack should not be empty.
4. After the push and pop operation print the stack elements
5. Stop the process.

PROGRAM:

```
class stack {
    private int arr[];
    private int top;
    private int capacity;

    stack(int size) {
        arr = new int[size];
        capacity = size;
        top = -1;
    }

    public void push(int x) {
        if (isFull()) {
            System.out.println("Stack Overflow");
            System.exit(1);
        }
        System.out.println("Inserting " + x);
        arr[++top] = x;
    }

    public int pop() {
        if (isEmpty()) {
            System.out.println("STACK EMPTY");
            System.exit(1);
        }
        return arr[top--];
    }

    public int getSize() {
        return top + 1;
    }
```

```
    public Boolean isEmpty() {
       return top == -1;
    }

    public Boolean isFull() {
       return top == capacity - 1;
    }

    public void printStack() {
       for (int i = 0; i <= top; i++) {
          System.out.print(arr[i] + " ");
       }
    }

    public static void main(String[] args) {
       stack stack = new stack(5);
       stack.push(1);
       stack.push(2);
       stack.push(3);
       System.out.print("Stack: ");
       stack.printStack();
       stack.pop();
       System.out.println("\nAfter popping out");
       stack.printStack();
    }
 }
```

OUTPUT:

```
Inserting 1
Inserting 2
Inserting 3
Stack: 1, 2, 3,
After popping out
1, 2
```

ExNo: 2(b) **QUEUE IMPLEMENTATION USING CLASS AND OBJECT**
Date:

AIM:
To develop Java program for Queue Implementation using class and object.

ALGORITHM:

1. **Enqueue**: Adds an item from the back of the queue.
2. **Dequeue**: Removes an item from the front of the queue.
3. **Front/Peek**: Returns the value of the item in front of the queue without dequeuing (removing) the item.
4. **IsEmpty**: Checks if the queue is empty.
5. **IsFull**: Checks if the queue is full.
6. **Display**: Prints all the items in the queue.

PROGRAM:

```
public class Queue {
   int SIZE = 5;
   int items[] = new int[SIZE];
   int front, rear;

   Queue() {
      front = -1;
      rear = -1;
   }

   boolean isFull() {
      if (front == 0 && rear == SIZE - 1) {
         return true;
      }
      return false;
   }

   boolean isEmpty() {
      if (front == -1) return true;
      else return false;
   }

   void enQueue(int element) {
      if (isFull()) {
         System.out.println("Queue is full");
      } else {
         if (front == -1) {
            front = 0;
         }
         rear++;
         items[rear] = element;
```

```
            System.out.println("Insert " + element);
        }
    }

    int deQueue() {
        int element;
        if (isEmpty()) {
            System.out.println("Queue is empty");
            return (-1);
        } else {
            element = items[front];
            if (front >= rear) {
                front = -1;
                rear = -1;
            } else {
                front++;
            }
            System.out.println(element + " Deleted");
            return (element);
        }
    }

    void display() {
        int i;
        if (isEmpty()) {
            System.out.println("Empty Queue");
        } else {
            System.out.println("\nFront index-> " + front);
            System.out.println("Items -> ");
            for (i = front; i <= rear; i++) System.out.print(items[i] + " ");
            System.out.println("\nRear index-> " + rear);
        }
    }

    public static void main(String[] args) {
        Queue q = new Queue();
        q.deQueue();
        for (int i = 1; i < 6; i++) {
            q.enQueue(i);
        }
        q.enQueue(6);
        q.display();
        q.deQueue();
        q.display();
    }
}
```

OUTPUT:

Rear index-> 4
1 Deleted

Front index-> 1
Items ->
2 3 4 5
Rear index-> 4

Sample Viva Questions:

1. What is meant by stack?
2. How Stack perform?
3. What are the operations done in stack?
4. What is LIFO and FIFO?
5. What are the operations done in queue?

Augmented Questions:

1. Write a Java Program on Write a Java Program to perform depth-first search (DFS) on a graph.
2. Write a Java Program to reverse a linked list

ExNo: 3 **GENERATING EMPLOYEE PAYROLL**
Date:

AIM:

To develop a java application for generating pay roll of employees with their Gross And Net salary.

ALGORITHM:

1. The package keyword is used to create a package in java.
2. Create a class Employee inside a package name employee.
3. Class Employee contains Emp_name, Emp_id, Address, Mail_id, Mobile_no as members.
4. By using Constructor initialize the instance variable of Employee class and display method is used to print employee details.
5. Create classes Programmer, AssistantProfessor, AssociateProfessor and Professor that extends Employee class and define necessary constructor for sub classes.
6. Each sub classes has its own instance variable like bPay and des.
7. Override the paySlip method in each sub classes to calculate the gross and net salary
8. By using super () method subclasses initialize the super class constructor.
9. Import employee package and create the object for Empolyee class.
10. Create different Employee object to add ArrayList<> classes.
11.DisplayEmployee method is used to display all employee playSlip detail

PROGRAM:

//For Packages, Folder Name should be ***employee***

//File Name should be ***Employee.java***

```
package employee;
public class Employee {
    private String name;
    private String id;
    private String address;
    private String mailId;
    private String mobileNo;
    public Employee(String name, String id, String address,
String mailId, String mobileNo) {
        this.name = name;
        this.id = id;
        this.address = address;
        this.mailId = mailId;
        this.mobileNo = mobileNo;
    }
public void display() {
        System.out.println("Emp_Name: " + name + "\t" + "Emp_id: " + id);
        System.out.println("Address: " + address);
        System.out.println("Mail_id: " + mailId + "\t" + "Mobile_no: " + mobileNo);
    }
```

```
    public void paySlip() {
        // Empty implementation
    }
}
```

*//For Packages, Folder Name should be **employee** //*

*File Name should be **Programmer.java***

```
package employee;

public class Programmer extends Employee {
    private float bPay;
    private String des;

    public Programmer(String name, String id, String address, String mailId, String mobileNo, float bPay,
String des) {
        super(name, id, address, mailId, mobileNo);
        this.bPay = bPay;
        this.des = des;
    }

    public void paySlip() {
        float da = bPay * 97 / 100;
        float hra = bPay * 10 / 100;
        double grossSalary = bPay + da + hra;
        float pf = bPay * 12 / 100;
        double scf = bPay * 0.1 / 100;
        double netSalary = grossSalary - pf - scf;
        System.out.println("------------ Employees Pay Slips ------------");
        super.display();
        System.out.println("Designation: " + des);
        System.out.println("Basic_Pay: " + bPay);
        System.out.println("Gross Salary: " + grossSalary + "\t" + "Net Salary: " + netSalary);
        System.out.println("------------ End of the Statements ------------");
    }
}
```

*//For Packages, Folder Name should be **employee***

*// File Name should be **AssistantProfessor.java***

```
package employee;

public class AssistantProfessor extends Employee {
    private float bPay;
    private String des;

    public AssistantProfessor(String name, String id, String address, String mailId, String mobileNo,
float bPay, String des) {
```

```
        super(name, id, address, mailId, mobileNo);
        this.bPay = bPay;
        this.des = des;
    }

    public void paySlip() {
        float da = bPay * 97 / 100;
        float hra = bPay * 10 / 100;
        double grossSalary = bPay + da + hra;
        float pf = bPay * 12 / 100;
        double scf = bPay * 0.1 / 100;
        double netSalary = grossSalary - pf - scf;
        System.out.println("------------ Employees Pay Slips ------------");
        super.display();
        System.out.println("Designation: " + des);
        System.out.println("Basic_Pay: " + bPay);
        System.out.println("Gross Salary: " + grossSalary + "\t" + "Net Salary: " + netSalary);
        System.out.println("------------ End of the Statements ------------");
    }
}
```

*//For Packages, Folder Name should be **employee***

*//File Name should be **AssociateProfessor.java***

```
package employee;

public class AssociateProfessor extends Employee {

    private float bPay;

    private String des;

    public AssociateProfessor(String name, String id, String address, String mailId, String mobileNo,
float bPay, String des) {

        super(name, id, address, mailId, mobileNo);

        this.bPay = bPay;

        this.des = des;

    }

    public void paySlip() {

        float da = bPay * 97 / 100;

        float hra = bPay * 10 / 100;

        double grossSalary = bPay + da + hra;

        float pf = bPay * 12 / 100;

        double scf = bPay * 0.1 / 100;
```

```
        double netSalary = grossSalary - pf - scf;
        System.out.println("------------ Employees Pay Slips ------------");
        super.display();
        System.out.println("Designation: " + des);
        System.out.println("Basic_Pay: " + bPay);
        System.out.println("Gross Salary: " + grossSalary + "\t" + "Net Salary: " + netSalary);
        System.out.println("------------ End of the Statements ------------");
    }
}
```

*//For Packages, Folder Name should be **employee***

*//File Name should be **Professor.java***

```
package employee;
public class Professor extends Employee {
    private float bPay;
    private String des;

    public Professor(String name, String id, String address, String mailId, String mobileNo, float bPay,
String des) {
        super(name, id, address, mailId, mobileNo);
        this.bPay = bPay;
        this.des = des;
    }
    public void paySlip() {
        float da = bPay * 97 / 100;
        float hra = bPay * 10 / 100;
        double grossSalary = bPay + da + hra;
        float pf = bPay * 12 / 100;
        double scf = bPay * 0.1 / 100;
        double netSalary = grossSalary - pf - scf;
        System.out.println("------------ Employees Pay Slips ------------");
        super.display();
        System.out.println("Designation: " + des);
```

```
            System.out.println("Basic_Pay: " + bPay);
            System.out.println("Gross Salary: " + grossSalary + "\t" + "Net Salary: " + netSalary);
            System.out.println("------------ End of the Statements ------------");
        }
    }
```

*//File Name should be **Emp.java** separate this file from above folder*

```
    import employee.*;
    import java.io.IOException;
    import java.util.ArrayList;
    import java.util.Scanner;

    public class Emp {
        Employee e;
        ArrayList<Employee> obj = new ArrayList<>();
        Scanner get = new Scanner(System.in);

        public void addEmployee() {
            System.out.println("Enter the Emp_Name:");
            String name = get.next();
            System.out.println("Enter the Emp_id:");
            String id = get.next();
            System.out.println("Enter the Address:");
            String address = get.next();
            System.out.println("Enter the Mail_id:");
            String mailId = get.next();
            System.out.println("Enter the Mobile_no:");
            String mobileNo = get.next();
            System.out.println("Enter the Designation:");
            String des = get.next();
            System.out.println("Enter the Basic_Pay:");
            float bPay = get.nextFloat();
```

```
        if (des.equalsIgnoreCase("Programmer")) {
            e = new Programmer(name, id, address, mailId, mobileNo, bPay, des);
            obj.add(e);
        } else if (des.equalsIgnoreCase("AssistantProfessor")) {
            e = new AssistantProfessor(name, id, address, mailId, mobileNo, bPay, des);
            obj.add(e);
        } else if (des.equalsIgnoreCase("AssociateProfessor")) {
            e = new AssociateProfessor(name, id, address, mailId, mobileNo, bPay, des);
            obj.add(e);
        } else if (des.equalsIgnoreCase("Professor")) {
            e = new Professor(name, id, address, mailId, mobileNo, bPay, des);
            obj.add(e);
        }
    }

    public void displayEmployee() {
        for (Employee e : obj) {
            e.paySlip();
        }
    }

    public static void main(String args[]) throws IOException {
        Emp x = new Emp();
        String check;
        do {
            x.addEmployee();
            System.out.println("Do you want to continue? Press 'y'");
            check = x.get.next();
        } while (check.equalsIgnoreCase("y"));
        x.displayEmployee();
    }
}
```

OUTPUT:

D:\>javac Emp.java

D:\>java Emp
Enter the Emp_Name: Suresh
Enter the Emp_id:E708
Enter the Address: cuddalore
Enter the Mail_id:suresh708@tgarments.org
Enter the Mobile_no: 7894561230
Enter the Designation:Programmer
Enter the Basic_Pay:7500
Do you wnat continue press 'y' y
Enter the Emp_Name: Rakesh
Enter the Emp_id:E705
Enter the Address: pondy
Enter the Mail_id: rakesh@gmail.com

Enter the Mobile_no: 4567891230
Enter the Designation:Professor
Enter the Basic_Pay:15000
Do you wnat continue press 'y' y
Enter the Emp_Name: kumar
Enter the Emp_id: E405
Enter the Address: madurai
Enter the Mail_id: kumarat@ymail.com
Enter the Mobile_no: 1237894560
Enter the Designation: AssistantProfessor
Enter the Basic_Pay:18000
Do you wnat continue press 'y' y
Enter the Emp_Name:Naresh
Enter the Emp_id: E102
Enter the Address:villupuram
Enter the Mail_id:nar12@rediffmail.co m
Enter the Mobile_no:9873214560
Enter the Designation:AssociateProfessor
Enter the Basic_Pay:20000
Do you wnat continue press 'y' n
---------- Employees Pay Slips
Emp_Name : Suresh

Emp_id : E708 Address : cuddalore

Mail_id : suresh708@tgarments.org

Mobile_no : 7894561230

Designation: Programmer

Basic_Pay: 7500.0

Gross Salary : 15525.0

Net Salary : 14617.5

---------- End of the Statements ----------

---------- Employees Pay Slips ----------

Emp_Name : Rakesh Emp_id : E705 Address :

pondy

Mail_id : rakesh@gmail.com Mobile_no : 4567891230

Designation: Professor

Basic_Pay: 15000.0

Gross Salary : 31050.0

Net Salary : 29235.0

---------- End of the Statements ----------

---------- Employees Pay Slips ----------

Emp_Name : kumar

Emp_id : E405 Address : madurai

Mail_id : kumarat@ymail.com Mobile_no : 1237894560

Designation: AssistantProfessor

Basic_Pay: 18000.0

Gross Salary : 37260.0

Net Salary : 35082.0

---------- End of the Statements ----------

---------- Employees Pay Slips ----------

Emp_Name : Naresh

Emp_id : E102

Address : villupuram

Mail_id : nar12@rediffmail.com Mobile_no : 9873214560

Designation: AssociateProfessor Basic_Pay: 20000.0

Gross Salary : 41400.0

Net Salary : 38980.0

---------- End of the Statements ----------

Sample Viva Questions:

1. What is Inheritance?
2. What are the type of inheritance?
3. What do you mean by access specifiers?
4. What type of inheritance not available in Java?
5. What benefits does inheritance offer in Java?

Augmented Questions:

1. Write a Java Program for Hotel booking system
2. Write a Java Program for banking system.
3. Implementing inheritance relationship for a car rental application using virtual lab.

ExNo: 4 FINDING THE AREA OF CIRCLE , RECTANGLE AND TRIANGLE USING ABSTRACT CLASS

Date:

AIM:

To write a java program to find the area of different shapes by using abstract class.

ALGORITHM:

1. Import the java packages.
2. Create an abstract class named Shape that contains two integers and an empty method named printArea().
3. Create a class Rectangle that extends the class Shape. Override the method printArea () by getting Width and Length then compute the area and prints the area of the Rectangle.
4. Create a class Triangle that extends the class Shape. Override the method printArea () by getting Base and Height then compute the area and prints the area of the Triangle.
5. Create a class Circle that extends the class Shape. Override the method printArea () by getting the Radius, then compute the area and prints the area of the Circle.
6. By using Scanner class get the input during runtime.
7. Create object for a class in memory and assign it to the reference variable, then the method is invoked.

PROGRAM:

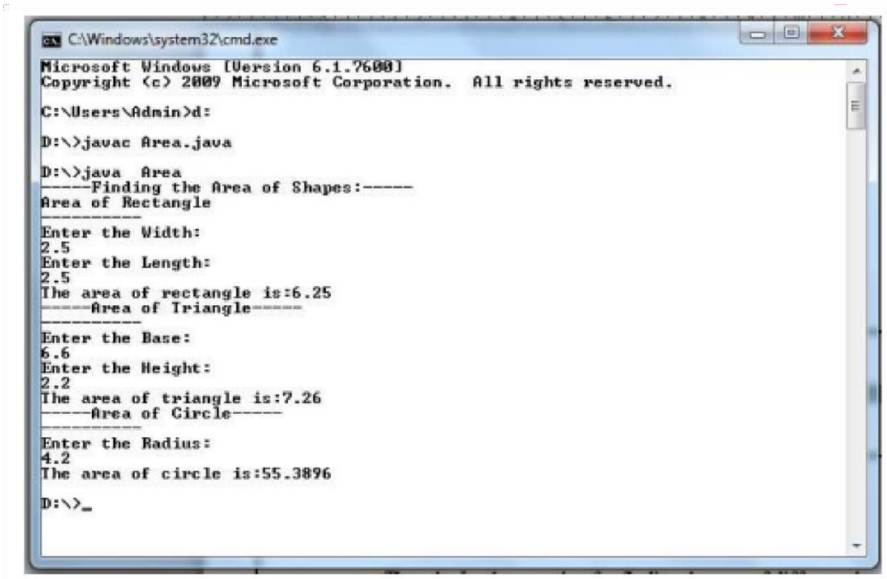
*//File Name should be **Area.java***

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

abstract class Shape {
   double a = 0.0, b = 0.0;
   abstract public void printArea();
}

class Rectangle extends Shape {
   double area = 0.0;

   public void printArea() {
      System.out.println("Area of Rectangle");
      System.out.println();
      Scanner in = new Scanner(System.in);
      System.out.println("Enter the Width:");
      this.a = in.nextDouble();
      System.out.println("Enter the Length:");
      this.b = in.nextDouble();
```

```
        this.area = a * b; /* (width * length) */
        System.out.println("The area of rectangle is:" + this.area);
    }
}

class Triangle extends Shape {
    double area = 0.0;

    public void printArea() {
        System.out.println("-----Area of Triangle");
        System.out.println();
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the Base:");
        this.a = in.nextDouble();
        System.out.println("Enter the Height:");
        this.b = in.nextDouble();
        this.area = 0.5 * a * b; /* 1/2 (base * height) */
        System.out.println("The area of triangle is:" + this.area);
    }
}

class Circle extends Shape {
    double area = 0.0;

    public void printArea() {
        System.out.println("-----Area of Circle");
        System.out.println();
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the Radius:");
        this.a = in.nextDouble();
        this.area = 3.14 * a * a;
        System.out.println("The area of circle is:" + this.area);
    }
}

public class Area {
    public static void main(String[] args) {
        System.out.println("-----Finding the Area of Shapes");
        Shape s;
        s = new Rectangle();
```

```
        s.printArea();
        s = new Triangle();
        s.printArea();
        s = new Circle();
        s.printArea();
    }
}
```

OUTPUT:



Sample Viva Questions:

1. What is an abstract class?
2. What is the difference between abstract class and concrete class?
3. When to use abstract class in Java?
4. Can an abstract class have constructor? Why?
5. What is Abstract method in Java?

Augmented Questions:

1. Write a Java program to create an abstract class Person with abstract methods eat() and exercise(). Create subclasses Athlete and LazyPerson that extend the Person class and implement the respective methods to describe how each person eats and exercises..
2. Write a Java program to create an abstract class Instrument with abstract methods play() and tune(). Create subclasses for Glockenspiel and Violin that extend the Instrument class and implement the respective methods to play and tune each instrument.

ExNo: 5 CIRCLE, RECTANGLE, TRIANGLE AREA CALCULATION USING AN INTERFACE
Date:

AIM:

To develop Java program for circle, rectangle ,triangle Area Calculation Using Interface.

ALGORITHM:

1. Import the java packages.
2. Create an Interface named Area that contains two integers and an method named Compute().
3. Create a class Rectangle that implements Area. then compute the area and prints the area of the Rectangle.
4. Create a class Triangle that implements the class Area. then compute the area and prints the area of theTriangle.
5. Create a class Circle that implenets the class Area. then compute the area and prints the area of the Circle.
6. Create object for a class in memory and assign it to the reference variable, then the method is invoked.

PROGRAM:

```
public interface Area {
   double Compute(double a, double b);
}

class Rectangle implements Area {
   public double Compute(double l, double b) {
      return (l * b);
   }
}

class Triangle implements Area {
   public double Compute(double b, double h) {
      return (b * h / 2);
   }
}

class Circle implements Area {
   public double Compute(double x, double y) {
      double pi = 3.14;
      return (pi * x * x);
   }
}

public class MainArea {
   public static void main(String args[]) {
      Rectangle rect = new Rectangle();
      double RArea = rect.Compute(10, 20);
      System.out.println("The area of the Rectangle is " + RArea);

      Triangle tri = new Triangle();
      double TArea = tri.Compute(10, 20);
      System.out.println("The area of the triangle is " + TArea);
```

```
        Circle cir = new Circle();
        double CArea = cir.Compute(15, 15);
        System.out.println("The area of the Circle is " + CArea);

}
```

OUTPUT:

```
The area of the Rectangle is 200.0
The area of the triangle is 100.0
The area of the Circle is 706.5
```

Sample Viva Questions:

1. What is an Interface in Java?
2. What is the significance of the default keyword in interface methods?
3. Can an interface extend another interface?
4. What is the difference between abstract classes and interfaces?
5. Can an interface extend multiple interfaces?

Augmented Questions:

1. Write a Java program to create an interface Searchable with a method search(String keyword) that searches for a given keyword in a text document. Create two classes Document and WebPage that implement the Searchable interface and provide their own implementations of the search() method".

2. Write a Java program to create an interface Resizable with methods resizeWidth(int width) and resizeHeight(int height) that allow an object to be resized. Create a class Rectangle that implements the Resizable interface and implements the resize methods.

ExNo: 6 **IMPLEMENTATION OF USER DEFINED EXCEPTIONS**
Date:

AIM:

To write a java program to implement user defined exception handling.

ALGORITHM:

1. Import the java packages.
2. Create a subclass of Exception named as MyException it has only a constructor plus an overloaded toString ( ) method that displays the value of the exception.
3. The exception is thrown when compute ( ) integer parameter is greater than 10.
4. The main ( ) method sets up an exception handler for MyException, then calls compute ( ) with a legal value (less than 10) and an illegal one to show both paths through the code.

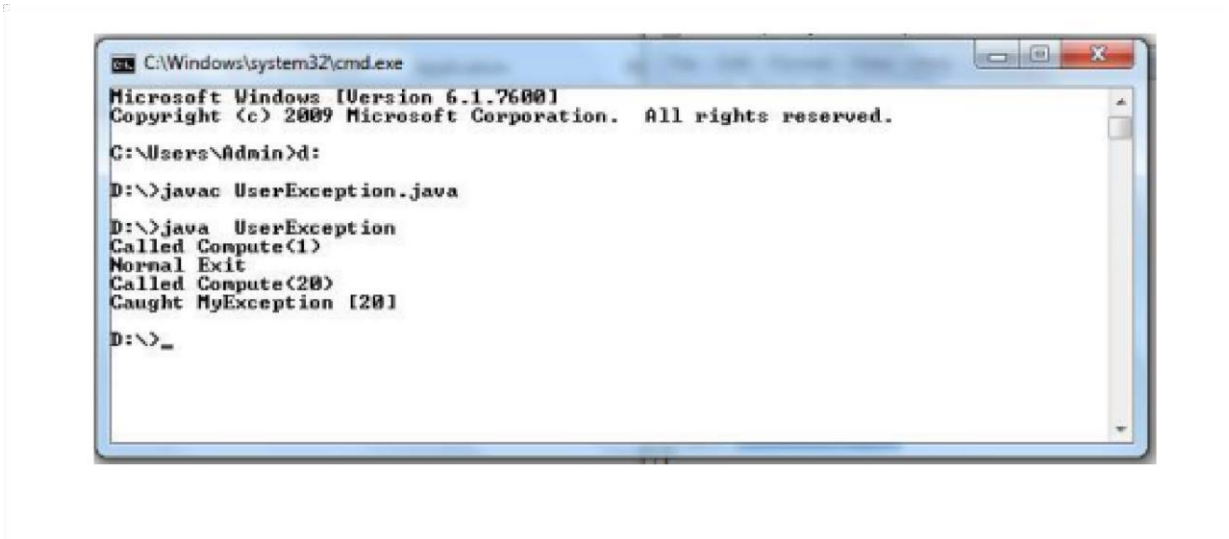
PROGRAM:

*//File Name should be **UserException.java***

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

class MyException extends Exception {
   private int d;

   MyException(int a) {
      d = a;
   }

   public String toString() {
      return "MyException [" + d + "]";
   }
}

class UserException {
   static void compute(int a) throws MyException {
      System.out.println("Called Compute(" + a + ")");
      if (a > 10) {
         throw new MyException(a);
      }
      System.out.println("Normal Exit");
   }
```

```
 public static void main(String args[]) {
       try {
          compute(1);
          compute(20);
       } catch (MyException e) {
          System.out.println("Caught " + e);
       }
    }
}
```

OUTPUT:



Sample Viva Questions:

1. What is an Exception?
2. What are the types of exception in Java?
3. List the exception handling keywords in Java?
4. What are built in exceptions?
5. Give the difference between error and exception inJ ava?

Augmented Questions:

1. Write a Java program that reads a list of numbers from a file and throws an exception if any of the numbers are positive.

2. Write a Java program to create a method that takes a string as input and throws an exception if the string does not contain vowel.

ExNo: 7 **MULTI THREADED APPLICATION**
Date:

AIM:

To write a program that implements a multi-threaded application.

ALGORITHM:

1. Import the java packages.
2. Create a thread that generates random number, Obtain one random number and check is odd or even.
3. If number is even then create and start thread that computes square of a number, Compute number * number and display the answer.
4. Notify to Random number thread and goto step 7.
5. If number is odd then create and start thread that computes cube of a number, Compute number * number * number and display the answer.
6. Notify to Random number thread and goto step 7.
7. Wait for 1 Second and Continue to Step 3 until user wants to exits.

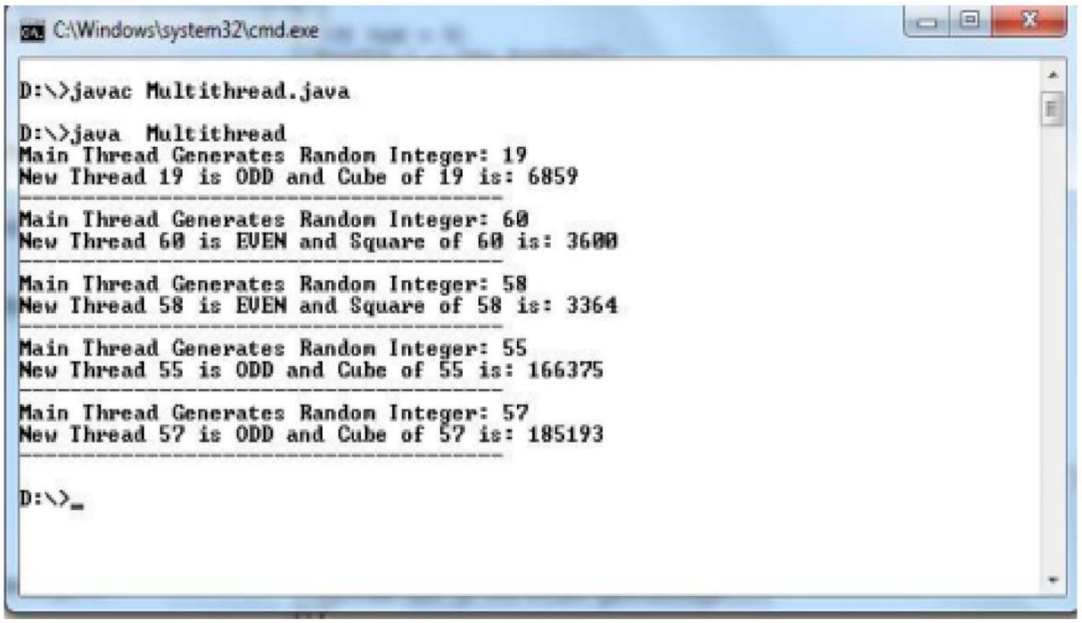
PROGRAM:

*//File Name should be **Multithread.java***

```
import java.util.*;

class Even implements Runnable {
   public int x;

   public Even(int x) {
      this.x = x;
   }

   public void run() {
      System.out.println("New Thread " + x + " is EVEN and Square of " + x + " is: " + x * x);
   }
}


class Odd implements Runnable {
   public int x;

   public Odd(int x) {
      this.x = x;
   }

   public void run() {
      System.out.println("New Thread " + x + " is ODD and Cube of " + x + " is: " + x * x * x);
   }
}
```

```
class Generate extends Thread {
   public void run() {
      int num = 0;
      Random r = new Random();

      try {
         for (int i = 0; i < 5; i++) {
            num = r.nextInt(100);
            System.out.println("Main Thread Generates Random
Integer: " + num);

            if (num % 2 == 0) {
               Thread t1 = new Thread(new Even(num));
               t1.start();
            } else {
               Thread t2 = new Thread(new Odd(num));
               t2.start();
            }

            Thread.sleep(1000);
            System.out.println(" ");
         }
      } catch (Exception ex) {
         System.out.println(ex.getMessage());
      }
   }
}

public class Multithread {
   public static void main(String[] args) {
      Generate g = new Generate();
      g.start();
   }
}
```

OUTPUT:



Sample Viva Questions:

1. What is multithreading?
2. What do you understand by inter-thread communication?
3. What are the advantages of multithreading?
4. What is the difference between wait() and sleep() method?
5. What does join() method?
6.

Augmented Questions:

1. Write a Java program to create and start multiple threads that decrement a shared counter variable concurrently.
2. Write a Java program that creates multiple threads to increment and decrement a shared bank account balance. Ensure the operations are thread-safe using synchronized methods or blocks.

ExNo: 8 **FILE OPERATIONS**
Date:

AIM:

To write a java program to implement file information such as reads a file name from the user, displays information about whether the file exists, whether the file is readable, or writable, the type of file and the length of the file in bytes.

ALGORITHM:

1. Import the java packages.
2. By using Scanner class get the input during runtime.
3. By using File class method create a File object associated with the file or directory specified by pathname. The pathname can contain path information as well as a file or directory name.
4. The exists() checks whether the file denoted by the pathname exists. Returns true if and only if the file denoted by the pathname exists; false otherwise
5. The getAbsolutePath() returns the absolute pathname string of the pathname.
6. The canRead() checks whether the application can read the file denoted by the pathname. Returns true if and only if the file specified by the pathname exists and can be read by the application; false otherwise.
7. The canWrite() checks whether the application can modify to the file denoted by the pathname. Returns true if and only if the file system actually contains a file denoted by the pathname and the application is allowed to write to the file; false otherwise.
8. The length() returns the length of the file denoted by the pathname. The return value is unspecified if the pathname denotes a directory.
9. The endsWith() returns true if the given string ends with the string given as argument for the method else it returns false.
10. The program uses conditional operator to check different functionalities of the given file.

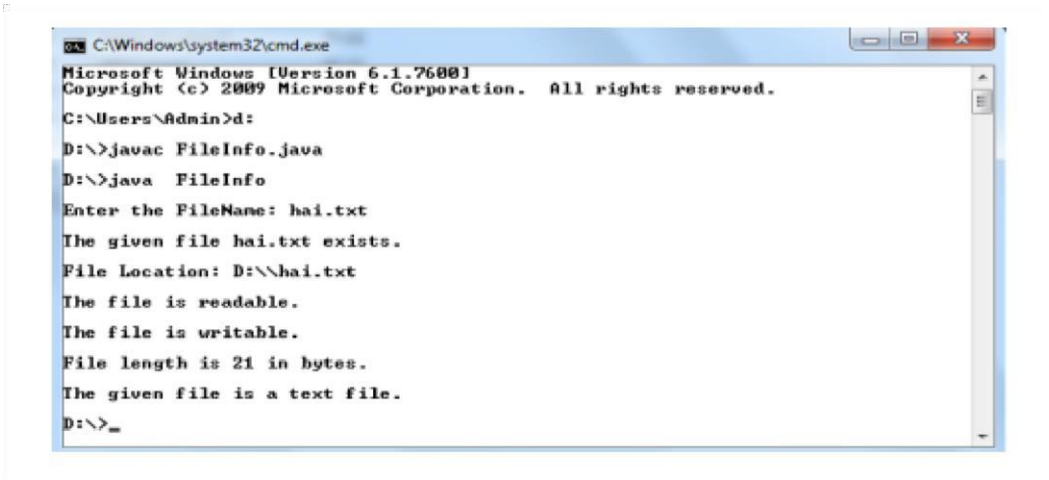
PROGRAM:

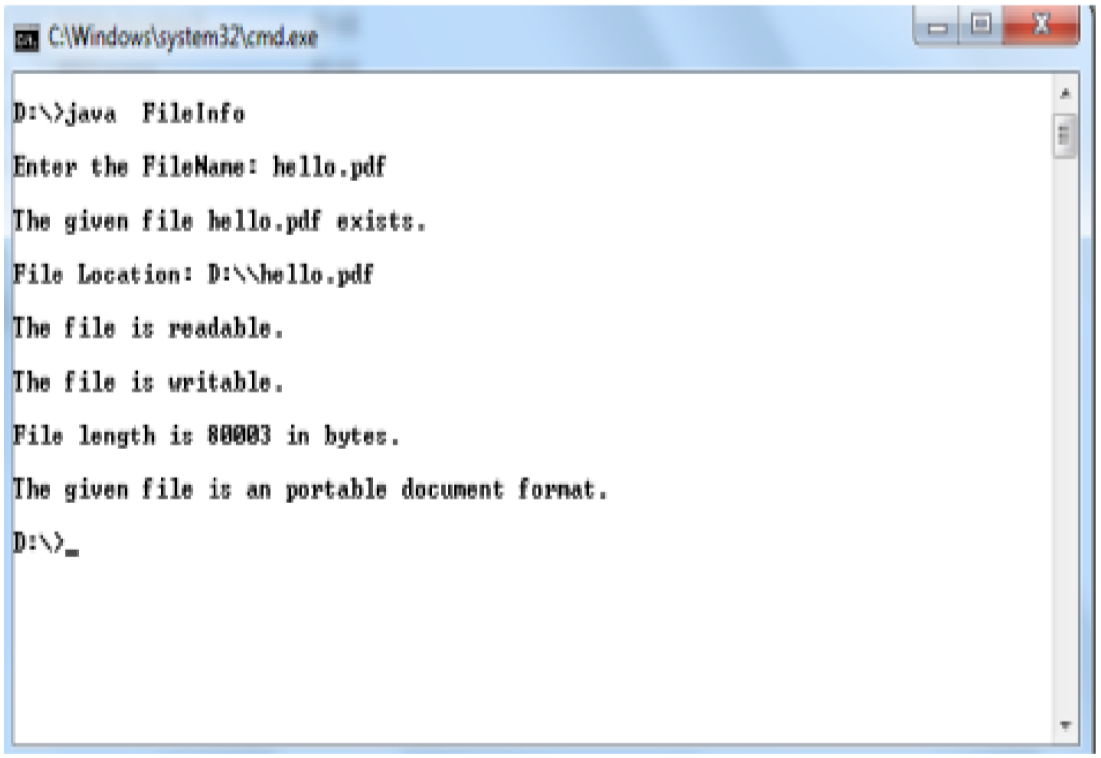
*//File Name should be **FileInfo.java***

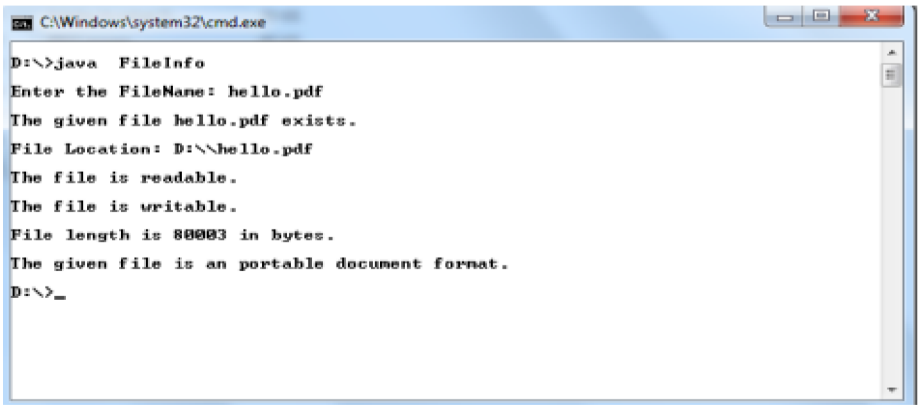
```
import java.io.*;
import java.util.*;

public class FileInfo {
   public static void main(String[] args) throws IOException {
      Scanner in = new Scanner(System.in);
      System.out.print("\nEnter the FileName: ");
      String fName = in.next();
      File f = new File(fName);
      String result = f.exists() ? " exists." : " does not exist.";
      System.out.println("\nThe given file " + fName + result);
      System.out.println("\nFile Location: " + f.getAbsolutePath());

      if (f.exists()) {
         result = f.canRead() ? "readable." : "not readable.";
         System.out.println("\nThe file is " + result);
         result = f.canWrite() ? "writable." : "not writable.";
```

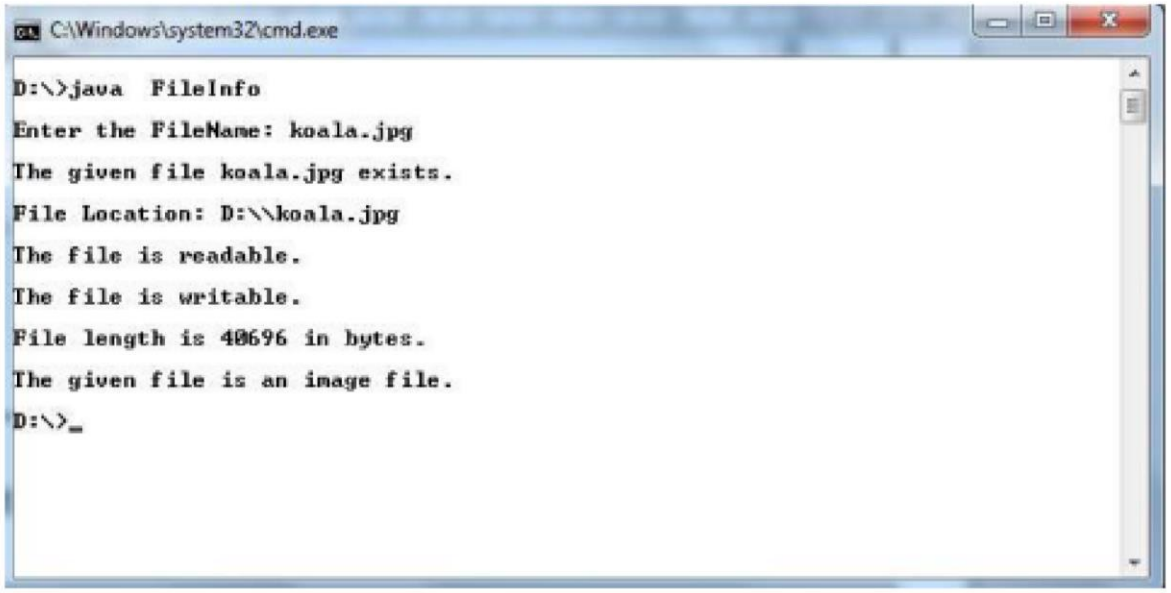
```
        System.out.println("\nThe file is " + result);
        System.out.println("\nFile length is " + f.length() + " in bytes.");

        if (fName.endsWith(".jpg") || fName.endsWith(".gif") || fName.endsWith(".png")) {
           System.out.println("\nThe given file is an image file.");
        } else if (fName.endsWith(".pdf")) {
           System.out.println("\nThe given file is a portable document format.");
        } else if (fName.endsWith(".txt")) {
           System.out.println("\nThe given file is a text file.");
        } else {
           System.out.println("The file type is unknown.");
        }
     }
  }
}
```

OUTPUT:




C:\Windows\system32\cmd.exe
D:\>java FileInfo
Enter the FileName: hello.pdf
The given file hello.pdf exists.
File Location: D:\\hello.pdf
The file is readable.
The file is writable.
File length is 80003 in bytes.
The given file is an portable document format.
D:\>


C:\Windows\system32\cmd.exe
D:\>java FileInfo
Enter the FileName: hello.pdf
The given file hello.pdf exists.
File Location: D:\\hello.pdf
The file is readable.
The file is writable.
File length is 80003 in bytes.
The given file is an portable document format.
D:\>



Sample Viva Questions:

1. What is file handling in Java, and why is it important in programming?
2. Explain the difference between text files and binary files.
3. What is the purpose of the 'File' class in Java, and how is it used for file handling?
4. What is the 'FileInputStream' class in Java, and how is it used to read binary files?
5. What is the 'FileWriter' class in Java, and how is it used for text file writing?

Augmented Questions:

1. Write a Java Program to merge the contents of two text files into a third file.
2. Write a Java program to count the frequency of each word in a text file and display the result.

ExNo: 9 FINDING THE MAXIMUM VALUE FROM THE GIVEN TYPE OF ELEMENTS USING GENERIC CLASS

Date:

AIM:

To write a java program to find the maximum value from the given type of elements using a generic classes..

ALGORITHM:

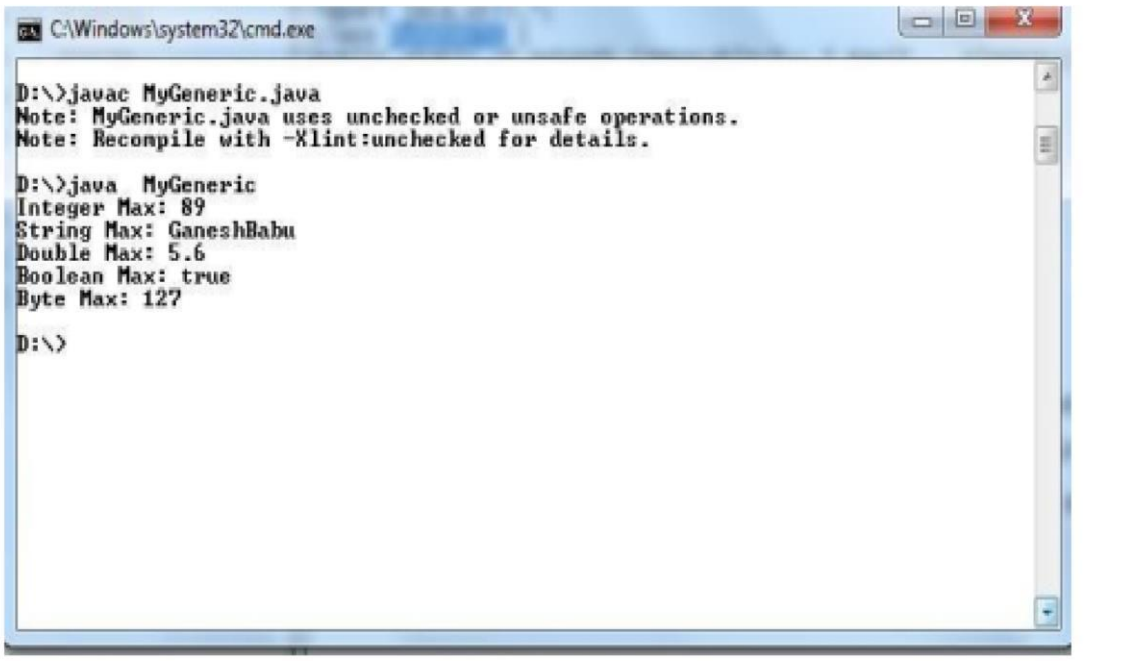
1. Import the java packages.
2. Comparable interface is used to order the objects of user-defined class.
3. This interface is found in java.lang package and contains only one method named compareTo(Object).
4. The compareTo() method works by returning an int value that is either positive, negative, or zero.
5. Create a generic method max(), that can accept any type of argument.
6. Then sets the first element as the max element, and then compares all other elements with the max element using compareTo() method
7. Finally the function returns an element which has the maximum value.
8. We can call generic method by passing with different types of arguments, the compiler handles each method.

PROGRAM:

*//File Name should be **MyGeneric.java***

```
import java.util.*;

class MyGeneric {
   public static <T extends Comparable<T>> T max(T... elements)
{
      T max = elements[0];
      for (T element : elements) {
         if (element.compareTo(max) > 0) {
            max = element;
         }
      }
      return max;
   }

   public static void main(String[] args) {
      System.out.println("Integer Max: " +
max(Integer.valueOf(32), Integer.valueOf(89)));
      System.out.println("String Max: " + max("GaneshBabu", "Ganesh"));
      System.out.println("Double Max: " + max(Double.valueOf(5.6), Double.valueOf(2.9)));
      System.out.println("Boolean Max: " + max(Boolean.TRUE, Boolean.FALSE));
      System.out.println("Byte Max: " + max(Byte.MIN_VALUE, Byte.MAX_VALUE));
   }
}
```

OUTPUT:



Sample Viva Questions:

1. What is Generics in Java?

2. What are the advantages of using Generics?

3. What are generic types?

4. What are generic methods?

5. What is type inference?

Augmented Questions:

1.Write a Java program to create a generic method that takes a list of numbers and returns the sum of all the positive and negative numbers.

2. Write a Java program to implement a generic method that finds the maximum element in an array of Comparable type..

ExNo: 10 IMPLEMENTATION OF JAVAFX CONTROLS, LAYOUTS AND MENUS
Date:

AIM:

To develop Java program for creating controls, layouts and menus using JavaFX.

ALGORITHM:

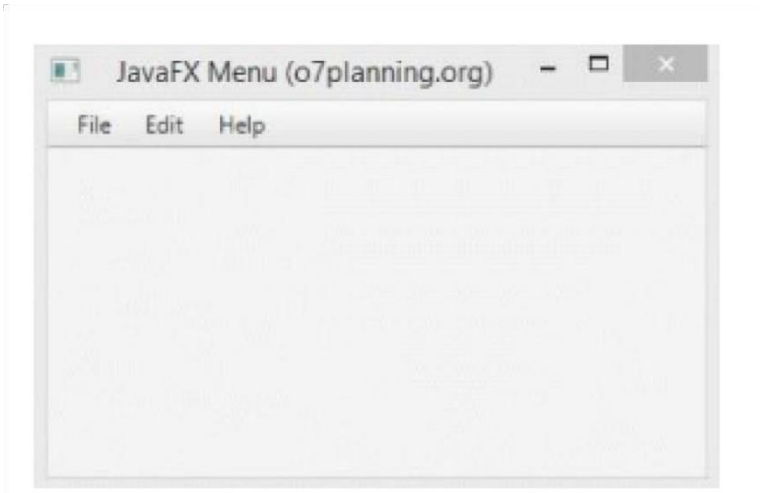
1. Open new JavaFX New Application and save file name as JavaFXMenuSample
2. Import Supporting packages into program and extends javafx application object Application.
3. Import menu package from javafx.scene.MenuBar.
4. Create menu and cerate menu items add the menu items to menu bar.
5. Launch the application and display the output.

PROGRAM:

```
package javafxapplicationmenu;

import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Menu;
import javafx.scene.control.MenuBar;
import javafx.scene.control.MenuItem;
import javafx.scene.layout.BorderPane;
import javafx.stage.Stage;

public class JavaFXApplicationMenu extends Application
{

   @Override
   public void start(Stage stage) {
      // Create MenuBar
      MenuBar menuBar = new MenuBar();

      // Create menus
      Menu fileMenu = new Menu("File");
      Menu editMenu = new Menu("Edit");
      Menu helpMenu = new Menu("Help");

      // Create MenuItems
      MenuItem newItem = new MenuItem("New");
      MenuItem openFileItem = new MenuItem("Open File");
      MenuItem exitItem = new MenuItem("Exit");
      MenuItem copyItem = new MenuItem("Copy");
      MenuItem pasteItem = new MenuItem("Paste");

      // Add menuItems to the Menus
      fileMenu.getItems().addAll(newItem, openFileItem, exitItem);
      editMenu.getItems().addAll(copyItem, pasteItem);
```

```
        // Add Menus to the MenuBar
        menuBar.getMenus().addAll(fileMenu, editMenu, helpMenu);

        BorderPane root = new BorderPane();
        root.setTop(menuBar);

        Scene scene = new Scene(root, 350, 200);
        stage.setTitle("JavaFX Menu (o7planning.org)");
        stage.setScene(scene);
        stage.show();
    }

    public static void main(String[] args) {
        Application.launch(args);
    }
}
```

OUTPUT:



Sample Viva Questions:

1. What is JavaFX?
2. What are the Different inbuilt layout panes?
3. What is the difference between a V box layout and an H box layout?
4. What is a layout?
5. What is the JavaFX scene builder?

Augmented Questions:

1. Write a JavaFX application that displays a list of items in a ListView. When an item is selected, show the selected item in a label below the list.
2. Write a JavaFX application with two RadioButton options ("Male" and "Female") in a ToggleGroup. Display the selected gender in a label when either option is selected.

ExNo: 11 **MINI PROJECT - OPAC SYSTEM FOR LIBRARY**

Date:

AIM:

To develop a mini project OPAC system for library using Java concepts.

ALGORITHM:

1. Import the awt,swing packages.
2. Extend the JFrame which implements action listener to the class datas.
3. Create the text field for id, name and button for next, address and the panel.
4. Create object for the get content pane().
5. Assign the length and breadth value for the layout using grid layout.
6. Add the new labels for ISBN and book name.
7. Add the new button for the next book
8. Create the book name under the driver jdbc odbc driver in the try block.
9. Create the object for exception as e and use it for catching the error.
10. Show all the records using show record.

PROGRAM:

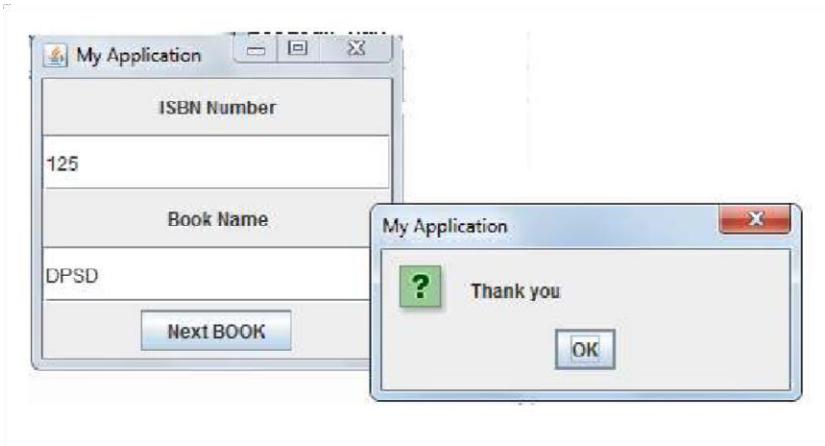
*//File Name should be **Data.java***

```
import java.sql.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Data extends JFrame implements ActionListener {

   JTextField id;
   JTextField name;
   JButton next;
   JButton addnew;
   JPanel p;
   static ResultSet res;
   static Connection conn;
   static Statement stat;

   public Data() {
      super("MyApplication");
      Container c = getContentPane();
      c.setLayout(new GridLayout(5, 1));
      id = new JTextField(20);
      name = new JTextField(20);
```

```
    next = new JButton("NextBOOK");
    p = new JPanel();

    c.add(new JLabel("ISBN Number", JLabel.CENTER));
    c.add(id);
    c.add(new JLabel("Book Name", JLabel.CENTER));
    c.add(name);
    c.add(p);

    p.add(next);
    next.addActionListener(this);

    pack();
    setVisible(true);
    addWindowListener(new WIN());
}

public static void main(String args[]) {
    Data d = new Data();
    try {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        conn = DriverManager.getConnection("jdbc:odbc:stu");
        // cust is the DSN Name
        stat = conn.createStatement();
        res = stat.executeQuery("Select * from stu");
        // stu is the table name
        res.next();
    } catch (Exception e) {
        System.out.println("Error" + e);
    }
    d.showRecord(res);
}

public void actionPerformed(ActionEvent e) {
    if (e.getSource() == next) {
        try {
            res.next();
        } catch (Exception e) {
            // Handle exception
        }
        showRecord(res);
```

```
        }
    }

    public void showRecord(ResultSet res) {
        try {
            id.setText(res.getString(2));
            name.setText(res.getString(3));
        } catch (Exception e) {
            // Handle exception
        }
    } // end of the main

    // Inner class WIN
    class WIN extends WindowAdapter {
        public void windowClosing(WindowEvent w) {
            JOptionPane jop = new JOptionPane();
            jop.showMessageDialog(null, "Thank you", "MyApplication",
JOptionPane.QUESTION_MESSAGE);
        }
    }
}
```

OUTPUT:



Sample Viva Questions:

1. What is OPAC?
2. What is the advantage of using OPAC?
3. Can we create student mark list preparation using this OPAC?
4. What do you mean by Database?
5. Is it mandatory to use database for any application? Justity?

Augmented Questions:

1. Write a JavaFX application with a CheckBox for enabling/disabling dark mode. When checked, change the background and text color of the scene accordingly.

2. Write a JavaFX application with a text input field and a button. When the button is clicked, display the text entered in the input field in a label.