PRATHYUSHA ENGINEERING COLLEGE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LAB MANUAL

for

CS3381-OBJECT ORIENTED PROGRAMMING LABORATORY

(Regulation 2021, III Semester)

(Odd Semester)

ACADEMIC YEAR: 2023 – 2024

PREPARED BY

B.Gunasundari,

Assistant Professor / CSE

PRATHYUSHA ENGINEERING COLLEGE

Exp. No.1

1.a) SEQUENTIAL SEARCH

Aim:

To write a java program to perform sequential search.

Algorithm:

```
Step 1: Set i to 1
Step 2: if i > n then go to step 7
Step 3: if A[i] = x then go to step 6
Step 4: Set i to i + 1
Step 5: Go to Step 2
Step 6: Print Element x Found at index i and go to step 8
Step 7: Print element not found
Step 8: Exit
```

Program:

```
public class SequentialSearch{

 public static void main(String[] args)
{
   int[] One = {2, 9, 6, 7, 4, 5, 3, 0, 1};
   int target = 4;
sequentialSearch(One, target);
}
 public static void sequentialSearch(int[] a, int b) {
   int index = -1;
   for (int i = 0; i < a.length; i++) {
     if (a[i] == b) {
       index = i;
       break;
     }
   }
   if (index == -1) {
     System.out.println("Target integer does not exist in the array");
   } else {
     System.out.println("Target integer is in index " + index + " of the array");
   }
 }
}
```

PRATHYUSHA ENGINEERING COLLEGE

Output:

```
Target integer is in index 4 of the array
```

PRATHYUSHA ENGINEERING COLLEGE

Result:

Thus the java program to perform sequential search is executed successfully and output is verified.

1.b) BINARY SEARCH

Aim:

To write a java program to perform binary search.

Algorithm:

```
Step 1: set beg = lower_bound, end = upper_bound, pos = - 1
Step 2: repeat steps 3 and 4 while beg <=end
Step 3: set mid = (beg + end)/2

Step 4: if a[mid] = val
        set pos = mid
        print pos
        go to step 6
        else if a[mid] > val
        set end = mid - 1
        else
        set beg = mid + 1
        [end of if]
        [end of loop]
Step 5: if pos = -1
        print "value is not present in the array"
        else:
        print "value is  present in  index  'pos'  of the array "
Step 6: exit
```

PRATHYUSHA ENGINEERING COLLEGE

Program:

```
class BinarySearch{
   public static int binarySearch(int arr[], int first, int last, int key){
      if (last>=first){
         int mid = first + (last - first)/2;
         if (arr[mid] == key){
         return mid;
         }
         if (arr[mid] > key){
         return binarySearch(arr, first, mid-1, key);//search in left subarray
```

```
        }else{
        return binarySearch(arr, mid+1, last, key);//search in right subarray
        }
      }
      return -1;
    }
    public static void main(String args[]){
      int arr[] = {10,20,30,40,50};
      int key = 30;
      int last=arr.length-1;
      int result = binarySearch(arr,0,last,key);
      if (result == -1)
        System.out.println("Element is not found!");
      else
        System.out.println("Element is found at index: "+result);
    }
}
```

Output:

```
Element is found at index: 2
```

Result:

Thus the java program to perform binary search is executed successfully and output is verified.

PRATHYUSHA ENGINEERING COLLEGE

1.c) Selection Sort

Aim:

To write a java program to perform selection sort.

Algorithm:

Step1: Initialize minimum value(**min_idx**) to location 0.
Step2:Traverse the array to find the minimum element in the array.
Step3:While traversing if any element smaller than **min_idx** is found then swap both the values.
Step4:Then, increment **min_idx** to point to the next element.
Step5:Repeat until the array is sorted.

PROGRAM:

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

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

      selectionSort(arr1);
```

PRATHYUSHA ENGINEERING COLLEGE

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

Output:

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

PRATHYUSHA ENGINEERING COLLEGE

Result:

Thus the java program to perform selection sort is executed successfully and output is verified.

1.d) INSERTION SORT

Aim:

To write a java program to perform insertion sort.

Algorithm:

Step 1 - If the element is the first element, assume that it is already sorted. Return 1.

Step2 - Pick the next element, and store it separately in a **key.**

Step3 - Now, compare the **key** with all elements in the sorted array.

Step 4 - If the element in the sorted array is smaller than the current element, then move to the next element. Else, shift greater elements in the array towards the right.

Step 5 - Insert the value.

Step 6 - Repeat until the array is sorted.

Program:

```
public class InsertionSort{
  public static void insertionSort(int array[]) {
    int n = array.length;
    for (int j = 1; j < n; j++) {
      int key = array[j];
      int i = j-1;
      while ( (i > -1) && ( array [i] > key ) ) {
        array [i+1] = array [i];
        i--;
      }
      array[i+1] = key;
    }
  }

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

PRATHYUSHA ENGINEERING COLLEGE

```
      System.out.println();

      insertionSort(arr1);//sorting array using insertion sort

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

Output:

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

PRATHYUSHA ENGINEERING COLLEGE

Result:

Thus the java program to perform insertion sort is executed successfully and output is verified.

Exp. No.2.a)

STACK IMPLEMENTATION

AIM:

To Develop stack data structure using classes and objects.

ALGORITHM:

```
    1) Start.
    2) Initialize stack with maximum size N, top = -1,N=20.
    3) Define a class that Implement  an interface stack.
    4) Define a method push.
    4.1) if (top+1>=size)   then throw an  overflow exception .
    4.2) else add an element to stack
    5) Define a method pop
        5.1) if(top==-1)  then throw underflow ecxeption.
        5.2) else pop element from stack
  6) Define a method to display the content of the stack.
  7)  Stop.
```

PROGRAM:

```
import java.util.Scanner;
class Stack
{
   int top;
   int maxsize = 10;
   int[] arr = new int[maxsize];


   boolean isEmpty()
   {
      return (top < 0);
   }
   Stack()
   {
      top = -1;
   }
   boolean push (Scanner sc)
   {
      if(top == maxsize-1)
      {
```

PRATHYUSHA ENGINEERING COLLEGE

```
      System.out.println("Overflow !!");
      return false;
    }
    else
    {
      System.out.println("Enter Value");
      int val = sc.nextInt();
      top++;
      arr[top]=val;
      System.out.println("Item pushed");
      return true;
    }
  }
  boolean pop ()
  {
    if (top == -1)
    {
      System.out.println("Underflow !!");
      return false;
    }
    else
    {
            System.out.println("Item popped "+arr[top]);

      top --;

      return true;
    }
  }
  void display ()
  {
    System.out.println("Printing stack elements .....");
    for(int i = top; i>=0;i--)
    {
      System.out.println(arr[i]);
    }
  }
}
public class Stack_Operations {
public static void main(String[] args) {
  int choice=0;
  Scanner sc = new Scanner(System.in);
```

PRATHYUSHA ENGINEERING COLLEGE

```
   Stack s = new Stack();
   while(choice != 4)
   {
      System.out.println("\nChoose one from the below options...\n");
      System.out.println("\n1.Push\n2.Pop\n3.Show\n4.Exit");
      System.out.println("\n Enter your choice \n");
      choice = sc.nextInt();
      switch(choice)
      {
         case 1:
         {
            s.push(sc);
            break;
         }
         case 2:
         {
            s.pop();
            break;
         }
         case 3:
         {
            s.display();
            break;
         }
         case 4:
         {
            System.out.println("Exiting....");
            System.exit(0);
            break;
         }
         default:
         {
            System.out.println("Please Enter valid choice ");
         }
      };
   }
}
}
```

PRATHYUSHA ENGINEERING COLLEGE

Output:

```
Choose one from the below options...


1.Push
2.Pop
3.Show
4.Exit

 Enter your choice

1
Enter Value
11
Item pushed

Choose one from the below options...


1.Push
2.Pop
3.Show
4.Exit

 Enter your choice

2
Item popped 22
Choose one from the below options...


1.Push
2.Pop
3.Show
4.Exit

 Enter your choice

2
Item popped 11
Choose one from the below options...
```

PRATHYUSHA ENGINEERING COLLEGE

```
1.Push
2.Pop
3.Show
4.Exit
Enter your choice

2
Underflow !!

Choose one from the below options...
1.Push
2.Pop
3.Show
4.Exit



 Enter your choice

4
```

PRATHYUSHA ENGINEERING COLLEGE

Result:

Thus the java program for stack implementation is executed successfully and output is verified

2. b) QUEUE:

Aim:

To implement queue data structure using class and object.

Algorithm:

1. Create an array arr of size **n** and
2. Take two variables **front** and **rear** both of which will be initialized to 0 which means the queue is currently empty.
3. ***Enqueue:*** *Addition of an element to the queue. Adding an element will be performed after checking whether the queue is full or not. If rear < n which indicates that the array is not full then store the element at arr[rear] and increment rear by 1 but if rear == n then it is said to be an Overflow condition as the array is full.*
4. ***Dequeue:*** *Removal of an element from the queue. An element can only be deleted when there is at least an element to delete i.e. rear > 0. Now, the element at arr[front] can be deleted but all the remaining elements have to shift to the left by one position in order for the dequeue operation to delete the second element from the left on another dequeue operation.*
5. ***Front:*** *Get the front element from the queue i.e. arr[front] if the queue is not empty.*
6. ***Display:*** *Print all elements of the queue. If the queue is non-empty, traverse and print all the elements from the index front to rear.*

Program:

```
class Queue {
  static private int front, rear, capacity;
  static private int queue[];

  Queue(int c)
  {
    front = rear = 0;
    capacity = c;
    queue = new int[capacity];
  }

  // function to insert an element
  // at the rear of the queue
  static void queueEnqueue(int data)
  {
    // check queue is full or not
    if (capacity == rear) {
      System.out.printf("\nQueue is full\n");
      return;
    }

    // insert element at the rear
    else {
      queue[rear] = data;
      rear++;
    }
    return;
  }
```

PRATHYUSHA ENGINEERING COLLEGE

```
// function to delete an element
// from the front of the queue
static void queueDequeue()
{
    // if queue is empty
    if (front == rear) {
        System.out.printf("\nQueue is empty\n");
        return;
    }

    // shift all the elements from index 2 till rear
    // to the right by one
    else {
        for (int i = 0; i < rear - 1; i++) {
            queue[i] = queue[i + 1];
        }

        // store 0 at rear indicating there's no element
        if (rear < capacity)
            queue[rear] = 0;

        // decrement rear
        rear--;
    }
    return;
}

// print queue elements
static void queueDisplay()
{
    int i;
    if (front == rear) {
        System.out.printf("\nQueue is Empty\n");
        return;
    }

    // traverse front to rear and print elements
    for (i = front; i < rear; i++) {
        System.out.printf(" %d <-- ", queue[i]);
    }
    return;
}

// print front of queue
static void queueFront()
{
    if (front == rear) {
        System.out.printf("\nQueue is Empty\n");
        return;
```

PRATHYUSHA ENGINEERING COLLEGE

```
    }
    System.out.printf("\nFront Element is: %d",
             queue[front]);
    return;
  }
}

public class StaticQueueinjava {

  // Driver code
  public static void main(String[] args)
  {
    // Create a queue of capacity 4
    Queue q = new Queue(4);

    // print Queue elements
    q.queueDisplay();

    // inserting elements in the queue
    q.queueEnqueue(20);
    q.queueEnqueue(30);
    q.queueEnqueue(40);
    q.queueEnqueue(50);

    // print Queue elements
    q.queueDisplay();

    // insert element in the queue
    q.queueEnqueue(60);

    // print Queue elements
    q.queueDisplay();

    q.queueDequeue();
    q.queueDequeue();
    System.out.printf(
      "\n\nafter two node deletion\n\n");

    // print Queue elements
    q.queueDisplay();

    // print front of the queue
    q.queueFront();
  }
}
```

PRATHYUSHA ENGINEERING COLLEGE

Output:

```
Queue is Empty
 20 <--  30 <--  40 <--  50 <--
Queue is full
 20 <--  30 <--  40 <--  50 <--

after two node deletion

 40 <--  50 <--
Front Element is: 40
```

PRATHYUSHA ENGINEERING COLLEGE

Result:

Thus the queue implementation using class and object is executed successfully and output verified

Exp.no:3

EMPLOYEE DETAILS

AIM:

To Develop a java application with 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 fund. Generate pay slips for the employees with their gross and net salary.

ALGORITHM:

1: Start.

2: Create a class employee with employee name,employee id,address,mail id,mobile num as members with in this employee class.

3: Define constructor for intializing the variables(employee name,employee id,address,mail id,mobile num).

4: Define method payslip with basicpay of double data type as parameter with in this method.

5.1: find dearness allowance(da), calculate da=basicpay*97/100.

5.2:find house rent allowance(hra),calculate hra=basicpay*10/100.

5.3: find provident fund (pf),calculate pf=basicpay*12/100.

5.4:find staff club fund ,calculate club=basicpay*0.1/100.

5.5: find gross salary,calculate gross_salary=basicpay+da+hra.

5.6: find net salary,calculate net_salary=(gross_salary)-(pf)-(club).

6.1:Create a class programmer and inherit the class from employee class add basicpay as the member of this inherited class.

6.2: Create a class assistant professer and inherit the class from employee class add basicpay as the member of this inherited calss.

6.3: Create a class associate professer and inherit the class from employee class add basicpay as the member of this inherited calss.

6.4:Create a class professer and inherit the class from employee class add basicpay as the member of this inherited calss.

7: declare a class employeedemo add main method in it.

8.1:create objects for all inherited classes and pass values as parameters.

8.2: Call the method payslip(basicpay) with the objects of inherited classes.

PRATHYUSHA ENGINEERING COLLEGE

9: Calculate and display the resullt.

10: End.

Program:

```
class employee
{
String empname, empid, address, mailid;
long mobileno;

employee (String name, String id, String addr, String mid,long mno)
{
empname=name;
empid=id;
address=addr;
mailid=mid;
mobileno=mno;
}
void payslip(double basicpay)
{
double da=basicpay*97/100;
double hra=basicpay*10/100;
double pf=basicpay*12/100;
double club=basicpay*0.1/100;
double gross_salary=basicpay+da+hra;
double net_salary=gross_salary-pf-club;
System.out.println("Name: "+empname);
System.out.println("Employee id: "+empid);
System.out.println("Basic pay: "+basicpay);
System.out.println("HRA: "+hra);
System.out.println("DA : "+da);
System.out.println("PF : "+pf);
System.out.println(" Staff club fund : "+club);
System.out.println("Gross salay ="+gross_salary);
System.out.println("Net salay ="+net_salary);
}

}

class programmer extends employee
{
double basicpay;
programmer(String name, String id, String addr, String mid,long mno,double bp)
{
super(name,id,addr,mid, mno);
basicpay=bp;
}
```

PRATHYUSHA ENGINEERING COLLEGE

```
}

class assistantprofessor extends employee
{
double basicpay;
assistantprofessor(String name, String id, String addr, String mid,long mno,double bp)
{
super(name,id,addr,mid, mno);
basicpay=bp;
}
}
class associateprofessor extends employee
{
double basicpay;
associateprofessor(String name, String id, String addr, String mid,long mno,double bp)
{
super(name,id,addr,mid, mno);
basicpay=bp;
}
}
class professor extends employee
{
double basicpay;
professor(String name, String id, String addr, String mid,long mno,double bp)
{
super(name,id,addr,mid, mno);
basicpay=bp;
}
}

class employeedemo
{
public static void main(String args[])
{
programmer p=new
programmer("Rani","p101","chennai","rani@gmail.com",9823211110L,22000);
assistantprofessor ap=new
assistantprofessor("Raji","ap101","chennai","raji@gmail.com",9232166101L,26000);
associateprofessor assop=new
associateprofessor("Ragu","assop101","chennai","ragu@gmail.com",9800210111L,28000);
professor prof=new
professor("Raja","prof101","chennai","raja@gmail.com",9003201111L,30000);
System.out.println("programmer");

p.payslip(p.basicpay);
System.out.println("Assistant professor");
ap.payslip(ap.basicpay);
System.out.println("Associate professor");
assop.payslip(assop.basicpay);
System.out.println("professor");
```

PRATHYUSHA ENGINEERING COLLEGE

```
prof.payslip(prof.basicpay);
}}
```

Output:

```
programmer
Name: Rani
Employee id: p101
Basic pay: 22000.0
HRA: 2200.0
DA : 21340.0
PF : 2640.0
 Staff club fund : 22.0
Gross salay =45540.0
Net salay =42878.0
Assistant professor
Name: Raji
Employee id: ap101
Basic pay: 26000.0
HRA: 2600.0
DA : 25220.0
PF : 3120.0
 Staff club fund : 26.0
Gross salay =53820.0
Net salay =50674.0
Associate professor
Name: Ragu
Employee id: assop101
Basic pay: 28000.0
HRA: 2800.0
DA : 27160.0
PF : 3360.0
 Staff club fund : 28.0
Gross salay =57960.0
Net salay =54572.0
professor
Name: Raja
Employee id: prof101
Basic pay: 30000.0
HRA: 3000.0
DA : 29100.0
PF : 3600.0
 Staff club fund : 30.0
Gross salay =62100.0
Net salay =58470.0
```

PRATHYUSHA ENGINEERING COLLEGE

PRATHYUSHA ENGINEERING COLLEGE

RESULT:

Thus the java application to generate payslips for the employees is executed successfully.

Exp. No.4.

ABSTRACT CLASS

AIM:

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

ALGORITHM:

```
1. Start.
2. Create an abstract class named shape that contains 2 integers, constructors & create an
   empty abstract  method named printarea().
3. Design  the class named rectangle that extends the class shape and implement the
   printarea()  method.
     3.1. area=dim1*dim2
     3.2. print the area of rectangle.
4. Design  the class named triangle that extends the class shape and implement the
   printarea()  method.
     4.1. area=(dim1*dim2)/2
     4.2. print the area of rectangle.
5. Design  the class named circle that extends the class shape and implement the
   printarea()  method.
     5.1. area=3.14* dim1*dim1
     5.2. print the area of rectangle.
6. Design a main class and create an object of three classes rectangle, triangle and circle.
   Through object call the printarea() method.
7. Stop.
```

Program:

```
abstract class shape
{
double dim1;
double dim2;
shape(double a) {
dim1 = a;
}

shape(double a, double b) {
dim1 = a;
dim2 = b;
}

abstract void printarea();
}

class Rectangle extends shape
```

PRATHYUSHA ENGINEERING COLLEGE

```
{
Rectangle(double a, double b) {
super(a, b);
}

void printarea() {
System.out.println(" Area for Rectangle="+(dim1*dim2));
}
}

class Triangle extends shape
{
Triangle(double a, double b) {
super(a, b);
}

void printarea() {

System.out.println(" Area for Triangle= "+(dim1 * dim2 / 2));

}

}
class Circle extends shape
{
Circle(double a, double b) {
super(a);
}

void printarea() {

System.out.println(" Area for circle= "+(3.14*dim1 * dim1));
}

}

class shapedemo
{

public static void main(String args[]) {


Rectangle r = new Rectangle(9, 5);
Triangle t = new Triangle(10, 8);
Circle c=new Circle(10,10);
shape sref;

sref = r;

sref.printarea();
```

PRATHYUSHA ENGINEERING COLLEGE

```
sref = t;

sref.printarea();
sref = c;

sref.printarea();

}

}
```

Output:

```
 Area for Rectangle=45.0
 Area for Triangle= 40.0
 Area for circle= 314.0
```

PRATHYUSHA ENGINEERING COLLEGE

RESULT:

Thus, the Java Program for finding area of Rectangle, Triangle and Circle using abstract method is executed successfully and output is verified.

Exp. No.4.

INTERFACE

AIM:

To Develop a Java Program to create an interface named Shape that contains an empty method named print Area(). Provide three classes named Rectangle, Triangle anda Circle such that each one of the classes implements the interface Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.

ALGORITHM:

1. Start.
2. Create an interfacenamed shape that an empty method named printarea().
3. Design the class named rectangle that implements the interface shape and implement the printarea() method.
    3.1. area=dim1*dim2
    3.2. print the area of rectangle.
4. Design the class named triangle that implements the interface shape and implement the printarea() method.
    4.1. area=(dim1*dim2)/2
    4.2. print the area of rectangle.
5. Design the class named circle that implements the interface shape and implement the printarea() method.
    5.1. area=3.14* dim1*dim1
    5.2. print the area of rectangle.
6. Design a main class and create an object of three classes rectangle, triangle and circle. Through object call the printarea() method.
7. Stop.

PRATHYUSHA ENGINEERING COLLEGE

Program:

```
interface shape
{

double PI=3.14;
public void printarea();
}

class Rectangle implements shape
{
   int dim1,dim2;
Rectangle(int a, int b) {
dim1=a;
```

```
dim2=b;ssss
}

public void printarea() {
System.out.println(" Area for Rectangle="+(dim1*dim2));
}
}

class Triangle implements shape
{
   int dim1,dim2;
Triangle(int a, int b) {
dim1=a;
dim2=b;
}

public void printarea() {

System.out.println(" Area for Triangle= "+(dim1 * dim2 / 2));

}

}
class Circle implements shape
{int dim1;
Circle(int a) {
dim1=a;
}

public void printarea() {

System.out.println(" Area for circle= "+(3.14*dim1 * dim1));
}

}

public class Interfacedemo
{

public static void main(String args[]) {

Rectangle r = new Rectangle(9, 5);
```

PRATHYUSHA ENGINEERING COLLEGE

```
Triangle t = new Triangle(10, 8);
Circle c=new Circle(10);
r.printarea();
t.printarea();
c.printarea();
}
}
```

Output:

```
Area for Rectangle=45
 Area for Triangle= 40
 Area for circle= 314.0
```

PRATHYUSHA ENGINEERING COLLEGE

Result:

Thus the Java Program to implements interface is executed successfully and output is verified.

Exp. No.:6

USER DEFINED EXCEPTION

AIM:

To develop a Java program to implement user defined exception handling.

ALGORITHM:

1) Start.
2) Create a class that represents user defined exception named as myexception which extends from Exception class.
3) Call constructor of parent exception within the class.
4) Create a main class that uses above myexception class.
5) Add main method with in the main class.
6) Throw an object of user defined exception with in the try block.
7) Create an catch block to caught an exception which is throw by try block.
8) Print the message from myexception object.
9) Display the result.
10) Stop.

PRATHYUSHA ENGINEERING COLLEGE

Program:

```
class MyException extends Exception
{
private int detail;
MyException(int a) {
detail = a;
}

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

class ExceptionDemo {
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:

```
Called compute(1)
Normal exit
Called compute(20)
Caught MyException[20]
```

PRATHYUSHA ENGINEERING COLLEGE

RESULT:

Thus , the java program for user defined exception handling is executed successfully and output is verified.

Exp. No.7

MULTITHREADED APPLICATION

AIM:

Develop 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, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.

ALGORITHM:

1)Start.
2) creating a class which is extending from ‘ Thread ‘ class.

3) In first thread class implement run() method

3.1) Initialize num value to zero.

3.2) creating object of an Random class as r.

3.3) num=r.nextInt(10)

It generates the random integer numbers from 0 to 9.

3.4) if the generated number is even number then create object of second thread class and start it.

3.5)else create object of third thread class and start it.

3.6) call Thread.sleep(1000)

4) In second thread class implement run() method

4.1)t print the square value of the given number.

5) In third thread class implement run() method

5.3)t print the cube value of the given number.

6) create main class and add main method with in that class.

6.1)creating the object of the first thread class and start it.

7)stop.

PRATHYUSHA ENGINEERING COLLEGE

Program:

```
import java.util.*;

class first extends Thread
{
public void run()
{
int num = 0;
Random r = new Random();
try
{
for (int i = 0; i < 5; i++)
{
num = r.nextInt(10);
System.out.println("First Thread and Generated Number is " + num);
 if (num % 2 == 0)
{
 even t1=new even(num);

t1.start();
}
else
 {

odd t2=new odd(num);

t2.start();
}

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

class even extends Thread
{
public int x;
public even(int a)
{
x = a;
}
public void run()
{
System.out.println("Second Thread "+ x +" is EVEN and Square of "
```

PRATHYUSHA ENGINEERING COLLEGE

```
                        + x + " is: " + x * x);
}
}



class odd extends Thread
{
public int x;
 public odd(int a)
{
x = a;
}
public void run()
{
System.out.println("Third Thread "+ x +" is ODD and Cube of "
                        + x + " is: " + x * x * x);
}
}


public class mthread
{
public static void main(String[] args)
{
first a = new first();
a.start();
}
}
```

Output:

```
First Thread and Generated Number is 8
Second Thread 8 is EVEN and Square of 8 is: 64
--------------------------------------
First Thread and Generated Number is 38
Second Thread 38 is EVEN and Square of 38 is: 1444
--------------------------------------
First Thread and Generated Number is 87
Third Thread 87 is ODD and Cube of 87 is: 658503
--------------------------------------
First Thread and Generated Number is 0
Second Thread 0 is EVEN and Square of 0 is: 0
--------------------------------------
First Thread and Generated Number is 83
Third Thread 83 is ODD and Cube of 83 is: 571787
--------------------------------------
```

PRATHYUSHA ENGINEERING COLLEGE

PRATHYUSHA ENGINEERING COLLEGE

Result:

Thus the java program that implements a multi-threaded application that has three threads have executed successfully and output is verified.

Exp. No.8:

FILE OPERATIONS

AIM:

To develop a Java program that 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) Start.
2) get the file name from user.
3) Check whether is exist or not. if the given file exist then,
3.1)  find whether the file is readable or not, then print the    result.
3.2) find whether the file is writeable or not, then print the result.
3.3) find the length of the file in  bytes.
3.4) check what is the extension of given file name.
a)if the file name ends with ".jpg",".png",".gif" extensions  then, print the
given file is an image file.
b) if the file name ends with ".exe" extension then,
print the given file is an executable file.
c)if the file name ends with ".java" extension then, print the given file is an
java file.
d)if the file name ends with ".txt" extension then,
print the given file is an text file.
e) if the file name is didn't mention with proper extensions at the end of the
file, Then, file is unknown.
6) Print the details of the file.
7) Exit.
```

Program:

```
import java.util.Scanner;
import java.io.File;
class filedemo
{
        public static void main(String[] args)
        {
                Scanner input=new Scanner(System.in);
                System.out.println("Enter file name");
                String fname=input.nextLine();
                File f = new File(fname);

      String result = f.exists() ? "exists." : "does not exist.";
      System.out.println("\nThe given file " + result);
      if(f.exists())
```

PRATHYUSHA ENGINEERING COLLEGE

```
    {
      result = f.canRead() ? "readable." : "not readable.";
      System.out.println("The given file is " + result);

      result = f.canWrite() ? "writable." : "not writable.";
      System.out.println("The given file is " + result);

      System.out.println("The given file length is " + f.length() + " in bytes.");


      if (fname.endsWith(".jpg") || fname.endsWith(".gif") || fname.endsWith(".png"))
      {
        System.out.println("The given file is an image file.");
      }
      else if (fname.endsWith(".exe"))
      {
        System.out.println("The given file is an executable file.");
      }
      else if (fname.endsWith(".txt"))
      {
        System.out.println("The given file is a text file.");
      }
       else if (fname.endsWith(".java"))
      {
        System.out.println("The given file is a java file.");
      }

      else
      {
        System.out.println("The file type is unknown.");
      }
    }
  }
}
```

PRATHYUSHA ENGINEERING COLLEGE

Output:

```
Enter file name
stackdemo.java

The given file exists.
The given file is readable
The given file is writable
The given file length is 2930 in bytes
The given file is a java file
```

PRATHYUSHA ENGINEERING COLLEGE

Result:

Thus the Java program for reading 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 have been executed successfully and output is verified.
Exp. No.9

GENERIC CLASS

AIM:

To develop an applications to demonstrate the features of generics classes.

ALGORITHM:

1) Start.

2) Create a generic class with Two type parameters.

3) Create method print() which displays the value of type parameter.

4) Define another class and include main method.

5) Create an object of generic class.

6) Call the print method.

7) stop.

PRATHYUSHA ENGINEERING COLLEGE

Program:

```
class  generic<T, U>
{
   T var1;
   U var2;
   generic (T a, U b)
   {
      var1 = a;
      var2 = b;
   }
       public void print()
   {
      System.out.println(var1);
      System.out.println(var2);
   }
}
 public class genericdemo
{
   public static void main (String[] args)
   {
      generic <String, Integer> g1 =new generic <String, Integer>("Java", 15);

      g1.print ();
```

```
    }
}
```

Output:

```
Java
15
```

PRATHYUSHA ENGINEERING COLLEGE

Result:

Thus an applications to demonstrate the features of generics classes is developed and output is verified.

10. Develop applications using JavaFX controls, layouts and menus.

Exp. No.:10 . a)

Registration Applications using JavaFX controls and layouts

Aim:

To Develop Registration application using JavaFX controls and layouts.

Algorithm:

1. Start
2. Import all packages
3. Create gridpane layout.
4. Create labels for name, gender, reservation, technologies known, educational qualification, and location
5. Use text for name.
6. Use toggle button for gender and reservation.
7. Use check box for educational qualification.
8. Use choice box for location.
9. Create button for register.
10. Add all controls to gridpane layout. Add layout to scene. Add scene to stage.
11. Stop.

Program:

```
import javafx.application.Application;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;

import javafx.geometry.Insets;
import javafx.geometry.Pos;

import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.CheckBox;
import javafx.scene.control.ChoiceBox;
import javafx.scene.control.ListView;
import javafx.scene.control.RadioButton;
import javafx.scene.layout.GridPane;
import javafx.scene.text.Text;
import javafx.scene.control.TextField;
import javafx.scene.control.ToggleGroup;
import javafx.scene.control.ToggleButton;
import javafx.stage.Stage;
```

PRATHYUSHA ENGINEERING COLLEGE

```
public class Registration extends Application {
  @Override
  public void start(Stage stage) {
    //Label for name
    Text nameLabel = new Text("Name");

    //Text field for name
    TextField nameText = new TextField();

    //Label for date of birth
    Text dobLabel = new Text("Date of birth");



    //Label for gender
    Text genderLabel = new Text("gender");

    //Toggle group of radio buttons
    ToggleGroup groupGender = new ToggleGroup();
    RadioButton maleRadio = new RadioButton("male");
    maleRadio.setToggleGroup(groupGender);
    RadioButton femaleRadio = new RadioButton("female");
    femaleRadio.setToggleGroup(groupGender);

    //Label for reservation
    Text reservationLabel = new Text("Reservation");

    //Toggle button for reservation
    ToggleButton Reservation = new ToggleButton();
    ToggleButton yes = new ToggleButton("Yes");
    ToggleButton no = new ToggleButton("No");
    ToggleGroup groupReservation = new ToggleGroup();
    yes.setToggleGroup(groupReservation);
    no.setToggleGroup(groupReservation);

    //Label for technologies known
    Text technologiesLabel = new Text("Technologies Known");

    //check box for education
    CheckBox javaCheckBox = new CheckBox("Java");
    javaCheckBox.setIndeterminate(false);
```

PRATHYUSHA ENGINEERING COLLEGE

```
//check box for education
CheckBox dotnetCheckBox = new CheckBox("DotNet");
javaCheckBox.setIndeterminate(false);

//Label for education
Text educationLabel = new Text("Educational qualification");

//list View for educational qualification
ObservableList<String> names = FXCollections.observableArrayList(
  "Engineering", "MCA", "MBA", "Graduation", "MTECH", "Mphil", "Phd");
ListView<String> educationListView = new ListView<String>(names);

//Label for location
Text locationLabel = new Text("location");

//Choice box for location
ChoiceBox locationchoiceBox = new ChoiceBox();
locationchoiceBox.getItems().addAll
  ("Hyderabad", "Chennai", "Delhi", "Mumbai", "Vishakhapatnam");

//Label for register
Button buttonRegister = new Button("Register");

//Creating a Grid Pane
GridPane gridPane = new GridPane();

//Setting size for the pane
gridPane.setMinSize(500, 500);

//Setting the padding
gridPane.setPadding(new Insets(10, 10, 10, 10));

//Setting the vertical and horizontal gaps between the columns
gridPane.setVgap(5);
gridPane.setHgap(5);

//Setting the Grid alignment
gridPane.setAlignment(Pos.CENTER);

//Arranging all the nodes in the grid
gridPane.add(nameLabel, 0, 0);
gridPane.add(nameText, 1, 0);
```

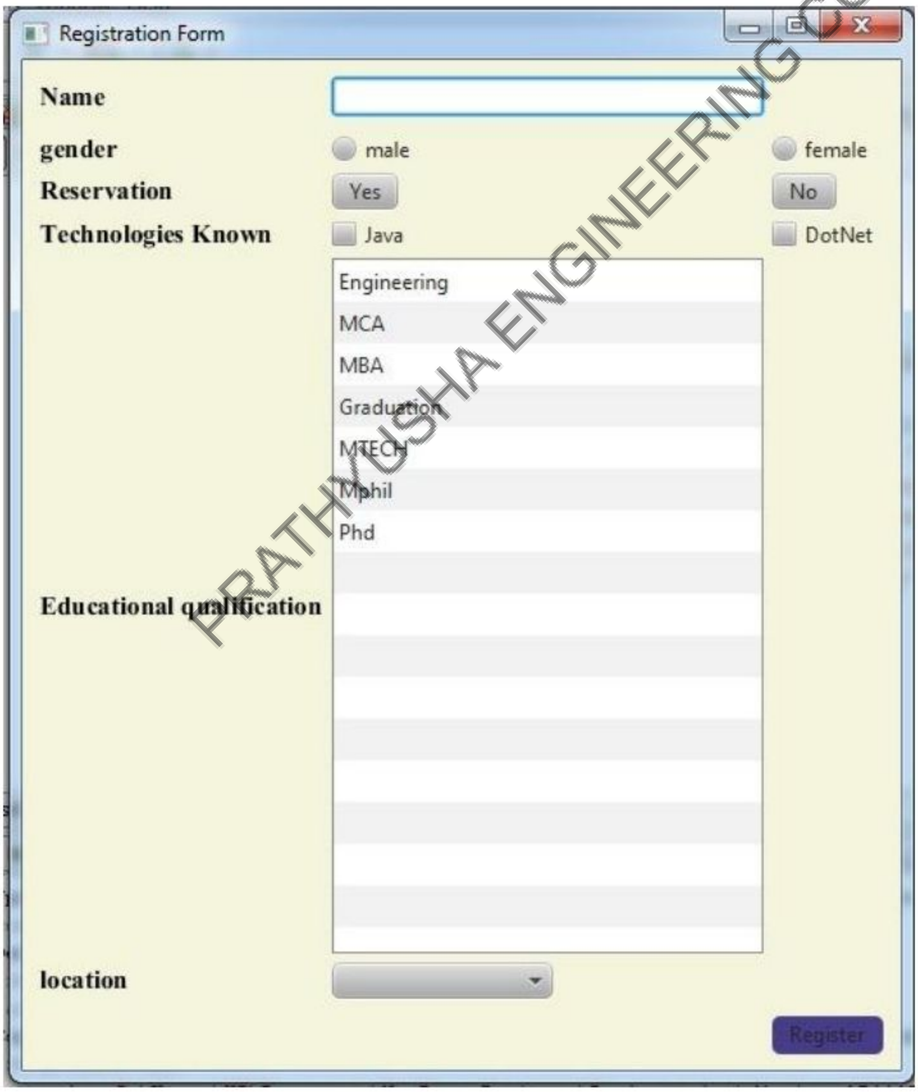
PRATHYUSHA ENGINEERING COLLEGE

```
    gridPane.add(dobLabel, 0, 1);

    gridPane.add(genderLabel, 0, 2);
    gridPane.add(maleRadio, 1, 2);
    gridPane.add(femaleRadio, 2, 2);
    gridPane.add(reservationLabel, 0, 3);
    gridPane.add(yes, 1, 3);
    gridPane.add(no, 2, 3);

    gridPane.add(technologiesLabel, 0, 4);
    gridPane.add(javaCheckBox, 1, 4);
    gridPane.add(dotnetCheckBox, 2, 4);

    gridPane.add(educationLabel, 0, 5);
    gridPane.add(educationListView, 1, 5);

    gridPane.add(locationLabel, 0, 6);
    gridPane.add(locationchoiceBox, 1, 6);

    gridPane.add(buttonRegister, 2, 8);

    //Styling nodes
    buttonRegister.setStyle(
      "-fx-background-color: darkslateblue; -fx-textfill: white;");

    nameLabel.setStyle("-fx-font: normal bold 15px 'serif' ");
    dobLabel.setStyle("-fx-font: normal bold 15px 'serif' ");
    genderLabel.setStyle("-fx-font: normal bold 15px 'serif' ");
    reservationLabel.setStyle("-fx-font: normal bold 15px 'serif' ");
    technologiesLabel.setStyle("-fx-font: normal bold 15px 'serif' ");
    educationLabel.setStyle("-fx-font: normal bold 15px 'serif' ");
    locationLabel.setStyle("-fx-font: normal bold 15px 'serif' ");

    //Setting the back ground color
    gridPane.setStyle("-fx-background-color: BEIGE;");

    //Creating a scene object
    Scene scene = new Scene(gridPane);


//Setting title to the Stage
```

PRATHYUSHA ENGINEERING COLLEGE

```
    stage.setTitle("Registration Form");

    //Adding scene to the stage
    stage.setScene(scene);

    //Displaying the contents of the stage
    stage.show();
  }
  public static void main(String args[]){
    launch(args);
  }
}
```

Output:



PRATHYUSHA ENGINEERING COLLEGE

PRATHYUSHA ENGINEERING COLLEGE

Result:

Thus the registration application using JavaFX controls and layouts is executed successfully and output is verified.

Exp. No.:10 .b)

Applications using JavaFX menu

Aim:

To Develop an applications using JavaFX menu.

Algorithm:

1. Start
2. Import all packages
3. Create borderpane layout.
4. Create menubar.
5. Create menu for file and edit. add menus to menu bar.
6. Create mennuitems new,save,exit and add to file menu.
7. Create menuitems cut,copy and paste and add to edit menu.
8. Add menubar to gridpane layout. Add layout to scene. Add scene to stage.
9. Stop.

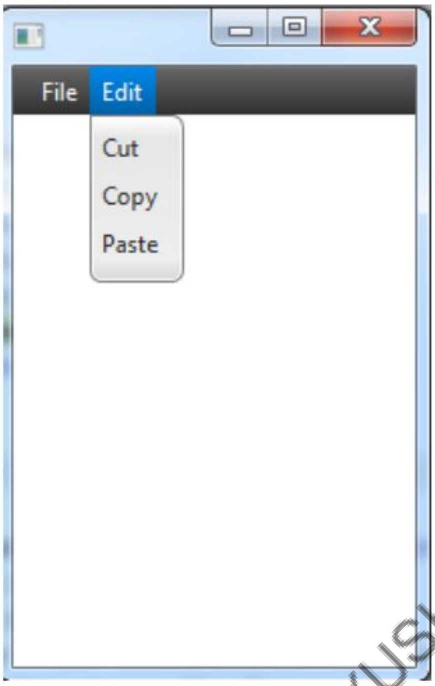
Program:

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.BorderPane;
import javafx.stage.Stage;
public class Menudemo extends Application {
  public static void main(String[] args) {
  launch(args);
      }

  @Override
  public void start(Stage primaryStage) throws Exception {
    // TODO Auto-generated method stub
    BorderPane root = new BorderPane();
    Scene scene = new Scene(root,200,300);
    MenuBar menubar = new MenuBar();
    Menu FileMenu = new Menu("File");
    MenuItem filemenu1=new MenuItem("new");
    MenuItem filemenu2=new MenuItem("Save");
    MenuItem filemenu3=new MenuItem("Exit");
    Menu EditMenu=new Menu("Edit");
    MenuItem EditMenu1=new MenuItem("Cut");
    MenuItem EditMenu2=new MenuItem("Copy");
    MenuItem EditMenu3=new MenuItem("Paste");
```

PRATHYUSHA ENGINEERING COLLEGE

```
        EditMenu.getItems().addAll(EditMenu1,EditMenu2,EditMenu3);
        root.setTop(menubar);
        FileMenu.getItems().addAll(filemenu1,filemenu2,filemenu3);
        menubar.getMenus().addAll(FileMenu,EditMenu);
        primaryStage.setScene(scene);
        primaryStage.show();

    }
}
```

Output:



PRATHYUSHA ENGINEERING COLLEGE

Result:

Thus an application using JavaFX menu is executed successfully and output is verified.
Exp. No.:11

MINI PROJECT

.

Write your project

PRATHYUSHA ENGINEERING COLLEGE

ADDITIONAL EXPERIMENTS

Exp. No.:12.

Javafx application to draw 2D shapes

Aim:

To develop javafx application to draw 2D shapes.

Algorithm:

1. Start
2. Create group.
3. Create circle by specifying center and radius.
4. Fill red color for circle.
5. Add group to scene. Add scene to stage.
6. Stop.

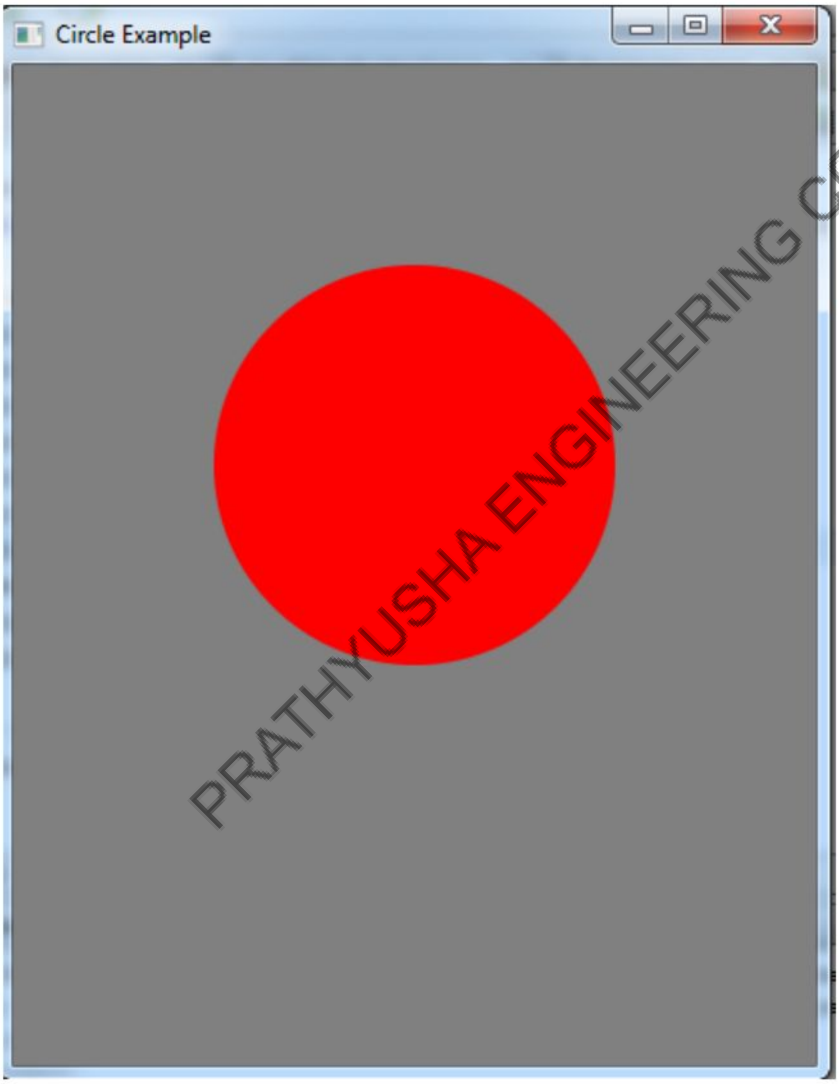
PRATHYUSHA ENGINEERING COLLEGE

Program:

```
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.stage.Stage;
public class Shape_Example extends Application{

  @Override
  public void start(Stage primaryStage) throws Exception {
    // TODO Auto-generated method stub
  primaryStage.setTitle("Circle Example");
  Group group = new Group();
  Circle circle = new Circle();
  circle.setCenterX(200);
  circle.setCenterY(200);
  circle.setRadius(100);
  circle.setFill(Color.RED);
  group.getChildren().addAll(circle);
```

```
        Scene scene = new Scene(group,400,500,Color.GRAY);
        primaryStage.setScene(scene);
        primaryStage.show();
    }
    public static void main(String[] args) {
        launch(args);
    }


    }
```

Output:



PRATHYUSHA ENGINEERING COLLEGE

Result:

Thus the javafx application to draw 2D shapes is successfully executed and output is verified.

Exp. No.:13.

CURRENCY CONVERTOR

AIM:

To develop a java application to implement currency converter using packages.

ALGORITHM:

1. Start

2. Declare a package for currency.

3. Declare a class “currencyconvert” with in the currency package.

4. Define the following methods.

    4.1. Read value to convert dollar to inr calculate rate=value*(66.66).

    4.2.Read value to convert euro to inr calculate rate=value*(79.84).

    4.3.Read value to convert yen to inr calculate rate=value*(0.61).

    4.4.Read value to convert inr to dollar calculate rate=value*(0.015).

    4.5.Read value to convert inr to euro calculate rate=value*(0.013).

    4.6.Read value to convert inr to yen calculate rate=value*(1.645).

5. Create another class and add main method & import currency package.

6. Create an object for currencyconvert class,

7. Using switch case , Call various method from currencyconvert class .

8. End.

PROGRAM:

```
package currency;


public class currencyconvert
{

 public double dollartoinr(double d)
{

return (d * 68.86);
}
```

PRATHYUSHA ENGINEERING COLLEGE

```
public double  eurotoinr(double d)
{
return (d * 78.14);
}
public double  yentoinr(double d)
{
return (d * 0.64);
}

public double inrtodollar(double d)
{
return (d * 0.015);
}
public double inrtoeuro(double d)
{
return (d * 0.013);
}
public double inrtoyen(double d)
{
return (d *1.57);
}

}



import currency.currencyconvert;
  import java.util.Scanner;


class currencydemo1
{
public static void main(String args[])
{

double dd;
   int    choice;

currencyconvert cc =new currencyconvert();

Scanner sc=new Scanner(System.in);

System.out.println( "1. Dollar to Inr 2. Euro to inr 3. Yen  to inr");
System.out.println( "4. Inr to dollar 5. inr  to Euro 6. Inr to yen   7.Exit");
do
{
   System.out.println( "Enter your choice : ");
  choice =sc.nextInt();
```

PRATHYUSHA ENGINEERING COLLEGE

```
   switch (choice)
   {
      case 1:
         System.out.println("Enter Dollar value ");
          dd=sc.nextDouble();
         System.out.println("Equivalent  INR is  "+cc.dollartoinr(dd));
         break;

case 2:
System.out.println("Enter Euro value");
          dd=sc.nextDouble();
         System.out.println("Equivalent  INR is  "+cc.eurotoinr(dd));
         break;

case 3:
System.out.println("Enter yen value");
          dd=sc.nextDouble();
         System.out.println("Equivalent  INR is  "+cc.yentoinr(dd));
         break;

case 4:
System.out.println("Enter int value");
          dd=sc.nextDouble();
         System.out.println("Equivalent  dollar is  "+cc.inrtodollar(dd));
         break;
case 5:
System.out.println("Enter inr value");
          dd=sc.nextDouble();
         System.out.println("Equivalent  euro is  "+cc.inrtoeuro(dd));
         break;
case 6:
System.out.println("Enter inr value");
          dd=sc.nextDouble();
         System.out.println("Equivalent  yen is  "+cc.inrtoyen(dd));
         break;

   }
      } while(choice!=7);
   }
}
```

PRATHYUSHA ENGINEERING COLLEGE

Output:

```
Z:\>javac -d . currencyconvert.java

Z:\>javac currencydemo1.java

Z:\>java currencydemo1
1. Dollar to Inr 2. Euro to inr 3. Yen  to inr
4. Inr to dollar 5. inr  to Euro 6. Inr to yen
Enter your choice :
1
Enter Dollar value
23
Equivalent  INR is  1583.78

Z:\>java currencydemo1
1. Dollar to Inr 2. Euro to inr 3. Yen  to inr
4. Inr to dollar 5. inr  to Euro 6. Inr to yen
Enter your choice :
1
Enter Dollar value
22
Equivalent  INR is  1514.92

Z:\>javac currencydemo1.java

Z:\>java currencydemo1
1. Dollar to Inr 2. Euro to inr 3. Yen  to inr
4. Inr to dollar 5. inr  to Euro 6. Inr to yen 7.Exit
Enter your choice :
1
Enter Dollar value
22
Equivalent  INR is  1514.92

Enter your choice :
2
Enter Euro value
34
Equivalent  INR is  2656.76
1. Dollar to Inr 2. Euro to inr 3. Yen  to inr
4. Inr to dollar 5. inr  to Euro 6. Inr to yen 7.Exit
Enter your choice :
3
Enter yen value
44
Equivalent  INR is  28.16
1. Dollar to Inr 2. Euro to inr 3. Yen  to inr
4. Inr to dollar 5. inr  to Euro 6. Inr to yen 7.Exit
Enter your choice : 7
```

PRATHYUSHA ENGINEERING COLLEGE

PRATHYUSHA ENGINEERING COLLEGE

RESULT:

The java application to implement currency convertor using package is executed successfully and output verified.