EX NO : 1 PROGRAM TO CHECK THE GIVEN STRING IS PALINDROM OR NOT

AIM

To write a Java program to find whether the given string is palindrome or not.

ALGORITHM

1. Start.
2. Read the input string from the user.
3. Remove spaces and convert the string to lowercase to make it case-insensitive.
4. Initialize two pointers, left and right, to the beginning and end of the string, respectively.
5. Repeat the following steps while left is less than right: a. Compare the character at position left with the character at position right. b. If they are not equal, return false (the string is not a palindrome). c. Increment left by 1. d. Decrement right by 1.
6. If the loop completes without returning false, return true (the string is a palindrome).
7. End.

PROGRAM

```
import java.util.Scanner;
public class PalindromeChecker
{
public static void main(String[] args)
{
Scanner scanner = new Scanner(System.in);
System.out.print("Enter a string: ");
String input = scanner.nextLine();
scanner.close();
if (isPalindrome(input))
{
System.out.println(input + " is a palindrome.");
}
Else
```

```
{
System.out.println(input + " is not a palindrome.");
}
}
public static boolean isPalindrome(String str) {
// Remove spaces and convert to lowercase to make the check case-insensitive
str = str.replaceAll("\\s+", "").toLowerCase();
 int  left = 0;
 int right = str.length() - 1;
 while (left < right) {
 if (str.charAt(left) != str.charAt(right))
 {
 return false;
 }
 left++;
 right--;
 }
 return true;
 }
 }
```

OUTPUT

Enter a string: hello

hello is not a palindrome.

Enter a string: A man a plan a canal Panama

A man a plan a canal Panama is a palindrome.

Enter a string: Was it a car or a cat I saw?

Was it a car or a cat I saw? is not a palindrome.

RESULT

Thus the java program verify the given string is palindrome or not.

EX NO : 2

PROGRAM TO GENERATE ELECTRICITY BILL

AIM

To develop a Java application to generate Electricity bill.

PROCEDURE

1.Create a class with the following members Consumer no., consumer name, previous month reading, current month reading, type of EB connection (i.e domestic or commercial)

2.Compute the bill amount using the following tariff.

If the type of the EB connection is domestic, calculate the amount to be paid as follows:

- First 100 units - Rs. 1 per unit
- 101-200 units - Rs. 2.50 per unit
- 201 -500 units - Rs. 4 per unit
- > 501 units - Rs. 6 per unit

If the type of the EB connection is commercial, calculate the amount to be paid as follows:

- First 100 units - Rs. 2 per unit
- 101-200 units - Rs. 4.50 per unit
- 201 -500 units - Rs. 6 per unit
- > 501 units - Rs. 7 per unit

3. Create the object for the created class .Invoke the methods to get the input from the consumer and display the consumer information with the generated electricity bill.

PROGRAM

```
import java.util.*;
class ebill
{
public static void main (String args[])
{
customerdata ob = new customerdata();
ob.getdata();
ob.calc();
ob.display();
}
}
class customerdata
{
Scanner in = new Scanner(System.in);
Scanner ins = new Scanner(System.in);
String cname,type;
int bn;
double current,previous,tbill,units;
void getdata()
{
System.out.print ("\n\t Enter consumer number  ");
bn = in.nextInt();
System.out.print ("\n\t Enter Type of connection  (D for Domestic or C for Commercial) ");
type = ins.nextLine();
System.out.print ("\n\t Enter consumer name  ");
cname = ins.nextLine();
System.out.print ("\n\t Enter previous month reading  ");
previous= in.nextDouble();
System.out.print ("\n\t Enter current month reading  ");
current= in.nextDouble();
}
void calc()
{
```

```
units=current-previous;
if(type.equals("D"))
{
if (units<=100)
tbill=1 * units;
else if (units>100 && units<=200)
tbill=2.50*units;
else if(units>200 && units<=500)
tbill= 4*units;
else
tbill= 6*units;
}
else
{
if (units<=100)
tbill= 2 * units;
else if(units>100 && units<=200)
tbill=4.50*units;
else if(units>200 && units<=500)
tbill= 6*units;
else
tbill= 7*units;
}
}
void display()
{
System.out.println("\n\t Consumer number = "+bn);
System.out.println ("\n\t Consumer name = "+cname);
if(type.equals("D"))
System.out.println ("\n\t type of connection  = DOMESTIC ");
else
System.out.println ("\n\t type of connection  = COMMERCIAL ");
System.out.println ("\n\t Current Month  Reading = "+current);
System.out.println ("\n\t Previous Month  Reading = "+previous);
System.out.println ("\n\t Total units = "+units);
```

```
System.out.println ("\n\t Total bill = RS "+tbill);
}
}
```

OUTPUT

```
Enter Consumer number : 6123
Enter the type of connection (D for Domestic or C for Commercial)D
Enter Consumer Name : ILAVARASI
Entger previous month reading 4800
Enter current month reading 5600
Consumer number = 6123
Enter Consumer Name : ILAVARASI
type of connection = Domestic
Current Month Reading= 5600.0
Previous Month Reading = 4800.0
Total units= 800.0
Total bill = Rs 4800.0
```

RESULT

Thus the java program to generate electricity bill was implemented and executed successfully.

EX NO: 3

PROGRAM TO IMPLEMENT CURRENCY CONVERTER USING PACKAGES

AIM

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

PROCEDURE

1. Create a Package currencyconversion and place the class currency under the package
2. Create the methods to perform currency conversion from dollar to rupee ,rupee to dollar,euro to rupee , rupee to euro , yen to rupee and rupee to yen.
3. Create the package distanceconverion and create the class distance within the package
4. Create the methods to convert from meter to km, km to meter, miles to km,km to miles
5. Create the package timeconversion and create the class timer .Create the methods to convert from hours to minutes ,hours to seconds , minutes to hours and seconds to hours
6. Create a class and import the packages currencyconversion,distanceconversion and time conversion.Create the objects for the class currency,distance and timer.
7. Get the choice from the user and invoke the methods to perform the corresponding conversion and display the value.

PROGRAM

currencyconversion.java

```
import java.util.*;
import java.text.DecimalFormat;
// currency converter in java swing
class CurrencyConverter
{
public static void main(String[] args
{
double rupee,dollar,pound,code,euro,yen;
DecimalFormat f = new DecimalFormat("##.###");
Scanner sc = new Scanner(System.in);
System.out.println("Enter the currency code :Rupees\n2:Dollar\n3:Pound\n4:Euro\n5:Yen");
code=sc.nextInt();
//For Rupees Conversion
if(code == 1)
{
System.out.println("Enter amount in rupees");
rupee = sc.nextFloat();
dollar = rupee / 66;
System.out.println("Dollar : "+f.format(dollar));
pound = rupee / 98;
System.out.println("Pound : "+f.format(pound));
euro = rupee / 72;
System.out.println("Euro : "+f.format(euro));
yen = rupee / 0.55;
System.out.println("Yen : "+f.format(yen));
}
//For Dollar Conversion
else if (code == 2)
{
System.out.println("Enter amount in dollar");
dollar = sc.nextFloat();
rupee = dollar * 66;
```

```
System.out.println("Rupees : "+f.format(rupee));
pound = dollar * 0.67;
System.out.println("Pound : "+f.format(pound));
euro = dollar * 0.92;
System.out.println("Euro : "+f.format(euro));
yen = dollar * 120.90;
System.out.println("Yen : "+f.format(yen));
}
 //For Pound Conversion
else if(code == 3)
{
 System.out.println("Enter amount in Pound");
 pound = sc.nextFloat();
 rupee = pound * 98;
 System.out.println("Rupees : "+f.format(rupee));
 dollar = pound * 1.49;
 System.out.println("Dollar : "+f.format(dollar));
 euro = pound * 1.36;
 System.out.println("Euro : "+f.format(euro));
 yen = pound * 179.89;
 System.out.println("Yen : "+f.format(yen));
 }
 //For Euro Conversion
 else if(code == 4)
 {
  System.out.println("Enter amount in Euro");
  euro = sc.nextFloat();
  rupee = euro * 72;
  System.out.println("Rupees : "+f.format(rupee));
  dollar = euro * 1.09;
  System.out.println("Dollar : "+f.format(dollar));
  pound = euro * 0.73;
  System.out.println("Pound : "+f.format(pound));
  yen = euro * 131.84;
  System.out.println("Yen : "+f.format(yen));
```

```
}
//For Yen Conversion
else if(code == 5)
{
System.out.println("Enter amount in Yen");
yen = sc.nextFloat();
rupee = yen * 0.55;
System.out.println("Rupees : "+f.format(rupee));
dollar = yen * 0.01;
System.out.println("Dollar : "+f.format(dollar));
pound = yen * 0.01;
System.out.println("Pound : "+f.format(pound));
euro = yen * 0.01;
System.out.println("Euro : "+f.format(euro));
}
else
System.out.println("Invalid Code");
}
}
```

OUTPUT

```
Enter the currency code
        1.Rupees
        2.Dollar
        3.Pound
        4.Euro
        5.Yen
1
Enter amount in rupees : 500
Dollar : 7.576
Pound :5.102
Euro : 6.944
Yen : 909.091
```

RESULT

Thus the java application to implement currency converter ,distance converter and time converter was implemented and executed successfully.

EX NO:4

PROGRAM FOR STACK ADT USING INTERFACE

AIM

To design a java application to implement array implementation of stack using the concept of interface and exception handling.

PROCEDURE

1. Create the interface stackoperation with method declarations for push and pop.
2. Create the class astack which implements the interface and provides implementation for the methods push and pop.Also define the method for displaying the values stored in the stack.Handle the stack overflow and stack underflow condition .
3. Create the class teststack .Get the choice from the user for the operation to be performed . and also handle the exception that occur while performing the stack operation.
4. Create the object and invoke the method for push,pop,display based on the input from the user.

PROGRAM

```
import java.io.*;
interface stackoperation
{
 public void push(int i),
 public void pop();
}
class Astack implements stackoperation
{
int stack[]=new int[5];
int top=-1;
int i;
public void push(int item)
{
if(top>=4)
{
System.out.println("overflow");
}
else
{
top=top+1;
stack[top]=item;
System.out.println("item pushed"+stack[top]);
}
}
public void pop()
{
if(top<0)
System.out.println("underflow");
else
{
System.out.println("item popped"+stack[top]);
top=top-1;

}
```

```
}
public void display()
{
if(top<0)
System.out.println("No Element in stack");
else
{
for(i=0;i<=top;i++)
System.out.println("element:"+stack[i]);
}
}
}
class teststack
{
public static void main(String args[])throws IOException
{
int ch,c;
int i;
Astack s=new Astack();
DataInputStream in=new DataInputStream(System.in);
do
{
try
{
System.out.println("ARRAY STACK");
System.out.println("1.push  2.pop  3.display  4.exit");
System.out.println("enter ur choice:");
ch=Integer.parseInt(in.readLine());
switch(ch)
{
case 1:
System.out.println("enter the value to push:");
```

```
i=Integer.parseInt(in.readLine());
s.push(i);
break;
case 2:
s.pop();
break;
case 3:
System.out.println("the elements are:");
s.display();
break;
case 4:
break;
}
}
catch(IOException e)
{
System.out.println("io error");
}
System.out.println("Do u want to continue 0 to quit and  1 to continue ");
c=Integer.parseInt(in.readLine());
}while(c==1);
}
}
```

OUTPUT

```
Array Stack
1.Push 2. Pop 3. Display 4.Exit
Enter your choice: 1
Enter the value to push: 10
Item pushed 10
Do you want to continue 0 to quit and 1 to continue
1
Array Stack
1.Push 2. Pop 3. Display 4.Exit
Enter your choice: 1
Enter the value to push: 10
Item pushed  20
Do you want to continue 0 to quit and 1 to continue
1
Array Stack
1.Push 2. Pop 3. Display 4.Exit
Enter your choice: 2
Enter the value to pop: 10
Item poped 10
Do you want to continue 0 to quit and 1 to continue
1
Array Stack
1.Push 2. Pop 3. Display 4.Exit
Enter your choice: 3
The elements are : 20
Do you want to continue 0 to quit and 1 to continue : 0
```

RESULT

Thus the java application for stack operations has been implemented and executed successfully.

EX NO: 5 PROGRAM TO PERFORM STRING OPERATIONS USING ARRAYLIST

AIM

To write a java program to perform string operations using ArrayList for the following functions

a. Append - add at end
b. Insert – add at particular index
c. Search

PROGRAM

```
import java.util.*;

import java.io.*;

public class arraylistexample

{

public static void main(String args[])throws IOException

{

ArrayList<String> obj = new ArrayList<String>();

DataInputStream in=new DataInputStream(System.in);

int c,ch;

int i,j;

String str,str1;

do

{

System.out.println("STRING MANIPULATION");

System.out.println("**********************************");

System.out.println(" 1. Append at end \t 2.Insert at particular index \t 3.Search \t");
```

```
System.out.println( "4.List string that starting with letter \t");
System.out.println("5.Size \t 6.Remove \t 7.Sort\t 8.Display\t" );
System.out.println("Enter the choice ");
c=Integer.parseInt(in.readLine());
switch(c)
{
case 1:
{
System.out.println("Enter the string ");
str=in.readLine();
obj.add(str);
break;
}
case 2:
{
System.out.println("Enter the string ");
str=in.readLine();
System.out.println("Specify the index/position to insert");
i=Integer.parseInt(in.readLine());
obj.add(i-1,str);
System.out.println("The array list has following elements:"+obj);
break;
}
```

```
case 3:
{
System.out.println("Enter the string to search ");
str=in.readLine();
j=obj.indexOf(str);
if(j==-1)
System.out.println("Element not found");
else
System.out.println("Index of:"+str+"is"+j);
break;
}
case 4:
{
System.out.println("Enter the character to List string that starts with specified character");
str=in.readLine();
for(i=0;i<(obj.size()-1);i++)
{
str1=obj.get(i);
if(str1.startsWith(str))
{
System.out.println(str1);
}
}
```

```
break;
}
case 5:
{
System.out.println("Size of the list "+obj.size());
break;
}
case 6:
{
System.out.println("Enter the element to remove");
str=in.readLine();
if(obj.remove(str))
{
System.out.println("Element Removed"+str);
}
else
{
 System.out.println("Element not present");
}
 break;
}
 case 7:
{
```

```
Collections.sort(obj);
System.out.println("The array list has following elements:"+obj);
break;
}
case 8:
{
System.out.println("The array list has following elements:"+obj);
break;
}
}
System.out.println("enter 0 to break and 1 to continue");
ch=Integer.parseInt(in.readLine());
}while(ch==1);
}
}
```

OUTPUT

String manipulation

1 Append at end 2 Insert at particular index 3 Search

4 List string with starting with letters 5 Size 6 Remove

7 Sort 8 Display

Enter the choice 1

Enter the string

One

Enter 0 to break and 1 to continue 1

String manipulation

1 Append at end 2 Insert at particular index 3 Search

4 List string with starting with letters 5 Size 6 Remove

7 Sort 8 Display

Enter the choice 2

Enter the string two

Speed the index / position to insert 2

The array list has following elements: one two

Enter 0 to break and 1 to continue 0

RESULT

Thus the java program to perform string operations using ArrayList has been implemented and executed successfully.

EX NO : 6 PROGRAM TO IMPLEMENT USER DEFINED EXCEPTION HANDLING

AIM

To write a java program to implement user defined exception handling

PROCEDURE

1.Create a class which extends Exception class.

2.Create a constructor which receives the string as argument.

3.Get the Amount as input from the user.

4.If the amount is negative , the exception will be generated.

5.Using the exception handling mechanism , the thrown exception is handled by the catch construct.

6.After the exception is handled , the string "invalid amount " will be displayed.

7.If the amount is greater than 0 , the message "Amount Deposited " will be displayed

PROGRAM

```
import java.util.Scanner;
class NegativeAmtException extends Exception
{
String msg;
NegativeAmtException(String msg)
{
this.msg=msg;
}
public String toString()
{
return msg;
}
}
public class userdefined
{
public static void main(String[] args)
{
Scanner s=new Scanner(System.in);
System.out.print("Enter Amount:");
int a=s.nextInt();
try
{
```

```
if(a<0)
{
throw new NegativeAmtException("Invalid Amount");
}
System.out.println("Amount Deposited");
}
catch(NegativeAmtException e)
{
System.out.println(e);
}
}
}
```

OUTPUT

```
c:\users\student\desktop(java user identified)

Enter Amount : 1000

Amount deposited

c:\users\student\desktop(java user identified)

Enter Amount : -1000

Invalid  Amount
```

RESULT

Thus a java program to implement user defined exception handling has been implemented and executed successfully.

EX NO :7 PROGRAM TO IMPLEMENT MULTITHREADED APPLICATION

AIM

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

PROCEDURE

1.Create a class even which implements first thread that computes .the square of the number .

2. run() method implements the code to be executed when thread gets executed.

3.Create a class odd which implements second thread that computes the cube of the number.

4.Create a third thread that generates random number.If the random number is even , it displays the square of the number.If the random number generated is odd , it displays the cube of the given number .

5.The Multithreading is performed and the task switched between multiple threads.

6.The sleep () method makes the thread to suspend for the specified time.

PROGRAM

```
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 A 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 and Generated Number is " + 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 multithreadprog
{
public static void main(String[] args)
{
A a = new A();
a.start();
}
}
```

OUTPUT

```
E:\Programs>java mulithreadprog

Main thread and Generated Number is 37

New Thread 37 is ODD and Cube of 37 is 50653

Main thread and Generated Number is 69

New Thread 37 is ODD and Cube of 37 is 328509
```

RESULT

Thus a java program for multi-threaded application has been implemented and executed successfully.

EX NO: 8 PROGRAM TO FIND THE MAXIMUM VALUE FROM THE GIVEN TYPE OF ELEMENTS USING GENERIC FUNCTION

AIM

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

PROCEDURE

1.Create a class Myclass to implement generic class and generic methods.

2.Get the set of the values belonging to specific data type.

3.Create the objects of the class to hold integer,character and double values.

4.Create the method to compare the values and find the maximum value stored in the array.

5.Invoke the method with integer, character or double values . The output will be displayed based on the data type passed to the method.

PROGRAM

```
class MyClass<T extends Comparable<T>>
{
T[] vals;
MyClass(T[] o)
{
vals = o;
}
public T min()
{
T v = vals[0];
for(int i=1; i < vals.length; i++)
if(vals[i].compareTo(v) < 0)
v = vals[i];
return v;
}
public T max()
{
T v = vals[0];
for(int i=1; i < vals.length;i++)
if(vals[i].compareTo(v) > 0)
v = vals[i];
return v;
```

```
}
}
class gendemo
{
public static void main(String args[])
{
int i;
Integer inums[]={10,2,5,4,6,1};
Character chs[]={'v','p','s','a','n','h'};
Double d[]={20.2,45.4,71.6,88.3,54.6,10.4};
MyClass<Integer> iob = new MyClass<Integer>(inums);
MyClass<Character> cob = new MyClass<Character>(chs);
MyClass<Double>dob = new MyClass<Double>(d);
System.out.println("Max value in inums: " + iob.max());
System.out.println("Min value in inums: " + iob.min());
System.out.println("Max value in chs: " + cob.max());
System.out.println("Min value in chs: " + cob.min());
System.out.println("Max value in chs: " + dob.max());
System.out.println("Min value in chs: " + dob.min());
}
}
```

OUTPUT

```
E:\Programs>Java gendemo
Max value in inums: 10
Min value in inums : 1
Max value in chs : V
Min value in chs : a
```

RESULT

Thus a java program to find the maximum value from the given type of elements has been implemented using generics and executed successfully.