UCS1312 OBJECT ORIENTED PROGRAMMING LABORATORY
LIST OF EXPERIMENTS

Ex.No.	Name of the Exercise	CO
1A	Simple Java program to display greetings message	CO1
1B	Java Program to Print Grade of a Student	CO1
2A	Java program to implement currency converter	CO1
2B	Java program to implement distance converter	CO1
2C	Java program to implement time converter	CO1
3	Java program to implement matrix multiplication using 2D arrays	CO1
4	Java program to implement string handling methods	CO2
5	Java program to implement constructors	CO3
6	Java program to manipulate Car price using class and objects	CO3
7	Java program to implement a Book class	CO3
8	Java program to implement Date class with exception handling	CO4
9	Java program to apply file handling methods	CO4
10	Java program to store and retrieve student details as structures in Files	CO4
11	Java program to implement interface using 2 classes UsedCar and SecSalesItem	CO5
CONTENT BEYOND SYLLABUS		
12	Java program to implement multi-threading application	CO3
13	Java program for Event driven application development	CO5

EX NO.1 SIMPLE JAVA PROG RAM TO PRINT A GREETING MESSAGE

Aim: To write a simple java program to print a greeting message

Algorithm:

1. Import the required packages.
2. Define the class MyFirstProgram
3. Get the *name* of the user using Scanner object
4. Display the message "Hai, <<*name*>> Welcome to My First Java Program"
   System.out.println("Hai, _____ Welcome to My First Java Program");
5. Save the program with the class name MyFirstProgram.java
6. Compile the program to generate the class file MyFirstProgram.class using the command
   >javac MyFirstProgram.java
7. Execute the program using the command
   >java MyFirstProgram

Program:

File name : Greet.java

```
import java.io.*;
import java.util.Scanner;
public class Greet
{
public static void main(String args[])
{
Scanner in=new Scanner(System.in);
System.out.println();
System.out.print("Please type your name here : ");
String name=in.nextLine();
System.out.println();
System.out.println("Hai "+name+", Welcome to My First Java Program !");
}
}
```

OUTPUT

```
D:\E DRIVE\OOP Java\LAB\UCS1312 LAB\JAVA PROGRAMS>javac Greet.java

D:\E DRIVE\OOP Java\LAB\UCS1312 LAB\JAVA PROGRAMS>java Greet

Please type your name here : James

Hai James, Welcome to My First Java Program !
```

EX NO.1B JAVA PROGRAM TO PRINT THE GRADE OF A STUDENT

AIM

To write a simple java program to find the grade of a student using fundamental programming structures in java.

ALGORITHM

1. Import the package java.io.*
2. Define a class Student with members name, m1,m2,m3,m4,m5 and Avg.
3. Get the name and marks for 5 subjects
4. Find the average marks for 5 subjects using the formula Avg=(m1+m2+m3+m4+m5)/5
5. Use else-if ladder structure to print the grade of the student.
6. Compile the program using

   ```
   >javac Grade .java
   ```

7. Execute the program to print the student details with grade using the command

   ```
   >java Grade
   ```

PROGRAM

File Name : Grade.java

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

class Grade
{
public static void main(String args[])
          {
            String name;
            int m1,m2,m3,m4,m5;
        float Avg;
        Scanner in=new Scanner(System.in);
        System.out.print(" Student Name  :  ");
            name=in.nextLine();
System.out.print(" Enter mark for subject1 :  ");
m1=in.nextInt();
System.out.print(" Enter mark for subject2 :  ");
m2=in.nextInt();
System.out.print(" Enter mark for subject3 :  ");
m3=in.nextInt();
System.out.print(" Enter mark for subject4 :  ");
m4=in.nextInt();
System.out.print(" Enter mark for subject5 :  ");
m5=in.nextInt();
            Avg=(m1+m2+m3+m4+m5)/5;
        if(Avg>95 &&Avg<=100)
                System.out.println("Grade of "+name+" is O");
        else if(Avg>90 &&Avg<=95)
                System.out.println("Grade of "+name+" is O");
        else if(Avg>80 &&Avg<=90)
                System.out.println("Grade of "+name+" is S");
        else if(Avg>70 &&Avg<=80)
                System.out.println("Grade of "+name+" is B");
        else if(Avg>60 &&Avg<=70)
                System.out.println("Grade of "+name+" is C");
        else if(Avg>50 &&Avg<=60)
                System.out.println("Grade of "+name+" is D");
        else
                System.out.println("Grade of "+name+" is U");
        }
}
```

OUTPUT

```
D:\JAVA>javac Grade.java

D:\JAVA>java Grade

Student Name  :  GOSLING

 Enter mark for subject1 :  98

 Enter mark for subject2 :  100

 Enter mark for subject3 :  80

 Enter mark for subject4 :  89

 Enter mark for subject5 :  91

Grade of GOSLING is O
```

EX.NO. 2a CURRENCY CONVERTER

AIM

To write a simple java program using class and object to perform conversions in INR, Yen, Dollar and Euro currencies.

ALGORITHM

1. Import the packages java.io.* and java.util.Scanner.
2. Define a class **CurrencyConv** with necessary 3 members Y, D and E
3. Define a constructor to initialize Y, D and E with Indian rupee equivalent of Yen, Dollar and Euro.

```
CurrencyConv()
{
        Y=0.57;
        D=83;
        E=91;
}
```

4. Define the method **toINR(yen,dollar,euro)** to find the INR equivalent using the formula

```
Rs_y = yen *0.57;
Rs_d = dollar * 83;
Rs_e = euro * 91;
```

5. Define the method **inrTo(inr)** to convert indian rupee to its equivalent yen dollar and euro.

```
y=inr/Y;
d= inr/D;
e=inr/E;
```

6. Define the main method and create object for *CurrencyConv* class using the statement, **CurrencyConv C=new CurrencyConv();**
7. Create a Scanner object *in* to read input from the user.
8. Define local variables **yen, euro, inr** and **dollar** as double datatype.
9. // "Currency Conversion (Yen, Dollar, Euro to INR)"
   i) Get the values of *yen, euro* and *dollar* using the scanner object and method **nextInt()**
   ii) Invoke the method **toINR(yen, dollar,euro)** to print the rupees equivalent.
10. // "Currency Conversion (INR to Yen, Dollar, Euro)"
   i) Get the values of *inr* using the scanner object and method **nextInt()**
   ii) Invoke the method **inrTo(inr)** to print the yen, dollar and euro equivalent.

// CURRENCY CONVERTER

PROGRAM

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

public class CurrencyConv
{

public static double yen,inr,euro,Y,D,E;

CurrencyConv()
{
Y=0.57;
D=83;
E=91;
}

public static void toINR(double yen,double dollar,double euro)
{
double rs_y,rs_d,rs_e;
rs_y= yen *  Y ;
rs_d= dollar * D;
rs_e= euro * E ;
System.out.println();
System.out.println("Rupees equivalent of " + yen + " Yen is \t :Rs."+ rs_y);
System.out.println("Rupees equivalent of " + dollar + " Dollar is \t :Rs."+ rs_d);
System.out.println("Rupees equivalent of " + euro + " Euro is \t :Rs."+ rs_e);
}

public static void inrTo(double inr)
{
double y,d,e;
y=inr/Y;
d= inr/D;
e=inr/E;
System.out.println();
System.out.println("Yen equivalent of " + inr +" rupees is \t:¥."+ String.format("%.2f",y));
System.out.println("Dollar equivalent of " + inr +" rupees is \t:$."+ String.format("%.2f",d));
System.out.println("Euro equivalent of " + inr +" rupees is \t:€."+String.format("%.2f",e));
}
public static void main(String args[])
{
CurrencyConv C=new CurrencyConv();
Scanner in=new Scanner(System.in);
double yen=0, dollar=0, euro = 0, i=0;
System.out.println("Currency Conversion (Yen, Dollar, Euro to INR)");
System.out.println("~~~~~~~~ ~~~~~~~~~~ ~~~~~~~~~~~~~~~~~~~~~~~~~~~");
System.out.print("Enter the YEN     :  ");
yen=in.nextInt();
System.out.print("Enter the DOLLAR  :  ");
dollar=in.nextInt();
System.out.print("Enter the EURO    :  ");
euro=in.nextInt();

toINR(yen, dollar,euro);

System.out.println("Currency Conversion (INR to Yen, Dollar, Euro)");
System.out.println("~~~~~~~~ ~~~~~~~~~~ ~~~~~~~~~~~~~~~~~~~~~~~~~~");
System.out.print("Enter the Indian Ruppes - INR  :  ");
i=in.nextInt();

inrTo(i);

}

}
```

OUTPUT

```
D:\Amudha\OOP JAVA 2023\UCS1312 LAB\JAVA PROGRAMS>javac CurrencyConv.java

D:\Amudha\OOP JAVA 2023\UCS1312 LAB\JAVA PROGRAMS>java CurrencyConv

Currency Conversion (Yen, Dollar, Euro to INR)
~~~~~~~~ ~~~~~~~~~~ ~~~~~~~~~~~~~~~~~
Enter the YEN    : 3
Enter the DOLLAR  : 5
Enter the EURO    : 2
Rupees equivalent of 3.0 Yen is        :Rs.1.71
Rupees equivalent of 5.0 Dollar is     :Rs.415.0
Rupees equivalent of 2.0 Euro is       :Rs.182.0

Currency Conversion (INR to Yen, Dollar, Euro)
~~~~~~~~ ~~~~~~~~~~ ~~~~~~~~~~~~~~~~~
Enter the Indian Ruppes - INR  :  100
Yen equivalent of 100.0 rupees is      :¥.175.44
Dollar equivalent of 100.0 rupees is     :$.1.20
Euro equivalent of 100.0 rupees is       :€.1.10
```

Result:

Thus the Java program to perform conversions between different types of currencies alike INR, Dollar and Euro was executed and verified.

EX.NO. 2b DISTANCE CONVERTER

AIM

To write a simple java program using class and object to convert meters and miles to kilometer and vice versa.

ALGORITHM

1. Import the packages java.io.* and java.util.Scanner.
2. Define a class **DistConv** with 2 members x and y.
3. Define a constructor **DistConv()** to initialize x and y.

   ```
   DistConv()
   {
         x=1000;        //1 km =1000 meter
         y=1.61;        //1 mile = 1.61 km
   }
   ```

4. Define the method **toKM(meter,miles)** to find the Kilometer equivalent using the formula

   ```
   km1=meter / x;
   km2=miles * y;
   ```

5. Define the method **kmTo(km)** to convert kilometer to its equivalent meters and miles.

   ```
   meters = km*x
   miles = km/y
   ```

6. Define the main method and create object for *DistConv* class using the statement,
   DistConv C=new DistConv();
7. Create a Scanner object *in* to read input from the user.
8. Define local variables **m1** and **m2** as double datatype to read meters and miles.
9. // "Distance Conversion (Meters and Miles to Kilometers)"
   i) Get the values of *m1*, and *m2* using the scanner object and method **nextInt()**
   ii) Invoke the method **toKM(m1,m2)** to print the kilometers equivalent.
10. // "Distance Conversion (KM to meters and miles)"
    i) Get the value of *km* using the scanner object and method **nextInt()**
    ii) Invoke the method **kmTo (km)** to print the meters and miles equivalent.

PROGRAM //DISTANCE CONVERTER

```
import java.io.*;
import java.util.Scanner;
public class DistConv
{
public static double x;
public static double y;
DistConv()
{
x=1000;        //1000 meter = 1 km
y=1.61;        //1 mile = 1.61 km
}

public static void toKM(double meter,double miles)
{
double km1,km2;
km1= meter/x;       // meter/1000
km2= miles*y ;      // miles*1.61

System.out.println();
System.out.println(meter + " meters is equivalent to "+km1+" Kilometers");
System.out.println(miles + " miles is equivalent to "+km2+" Kilometers");

}

public static void kmTo(double km)
{
System.out.println();
System.out.println(km + " kilometers is equivalent to "+ (km*x) +" meters");
System.out.println(km + " kilometers is equivalent to "+ (km/y) +" miles");
}

public static void main(String args[])
{
DistConv  D=new DistConv();
Scanner in=new Scanner(System.in);
double m1,m2,km;
System.out.println("Distance Conversion (meters and miles to Kilometers)");
System.out.println("~~~~~~~~~ ~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");
System.out.print("Enter the meters     :  ");
m1=in.nextInt();
System.out.print("Enter the miles :  ");
m2=in.nextInt();


toKM(m1,m2);

System.out.println("Distance Conversion (Kilometers to meters and miles)");
System.out.println("~~~~~~~~~ ~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");
System.out.print("Enter the Kilometers :  ");
km=in.nextInt();
kmTo(km);

}

}
```

OUTPUT

```
D:\Amudha\OOP JAVA 2023\UCS1312 LAB\JAVA PROGRAMS>javac
DistConv.java
D:\Amudha\OOP JAVA 2023\UCS1312 LAB\JAVA PROGRAMS>java DistConv
Distance Conversion (meters and miles to Kilometers)
~~~~~~~~ ~~~~~~~~~~ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Enter the meters    :  2000
Enter the miles :  100

2000.0 meters is equivalent to 2.0 Kilometers
100.0 miles is equivalent to 161.0 Kilometers
Distance Conversion (Kilometers to meters and miles)
~~~~~~~~ ~~~~~~~~~~ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Enter the Kilometers :  3

3.0 kilometers is equivalent to 3000.0 meters
3.0 kilometers is equivalent to 1.8633540372670807 miles
```

RESULT

Thus the Java program to perform conversions between different types of distance measure like meters, kilometers and miles was executed and verified.

EX.NO. 2c TIME CONVERTER

AIM

To write a simple java program using class and object to convert minutes and seconds to hours and vice versa.

ALGORITHM

```
1.  Import the packages java.io.* and java.util.Scanner.
2.  Create 2 classes MyTime and TimeConv
3.  Define class MyTime with constructor MyTime() to initialize hour, minute and second ans 0
              MyTime()
              {
                  h=0;
                  m=0;
                  s=0;
              }
4.  In class MyTime, define the method toHour() to find hour equivalent for the given minutes
    and seconds using the formula,
              hour1=minutes / 60;
              hour2=seconds / 360;
5.  In class MyTime, define the method hourTo() to convert hour to its equivalent minutes and
    seconds.
              minutes = hours * 60
              seconds = hours * 360
6.  In class TimeConv, define the main method and create object for MyTime class using the
    statement,
              MyTime D=new MyTime();
7.  Create a Scanner object in to read input from the user.
8.  //"Time Conversion (Minutes and seconds to hours)"
    Get the values of minutes and seconds using the scanner object in,
              D.m=in.nextInt();
              D.s=in.nextInt();
9.  Invoke the method toHour( ) to print the hours equivalent.
10. // "Time Conversion (Hours to minutes and Seconds)"
    Get the value of hours using the scanner object in,
              D.h=in.nextInt();
11. Invoke the method hourTo () to print the meters and miles equivalent.
12. Compile and Run the program
```

PROGRAM

```
import java.io.*;
import java.util.Scanner;
class MyTime
{
public static double h;
public static double m;
public static double s;

MyTime()
{
h=0;
m=0;
s=0;
}

public static void toHour()
{
System.out.println();
System.out.println(m + " Minutes is equivalent to  " + String.format("%.2f",m/60) +" Hours " );
System.out.println(s+ " Seconds is equivalent to  " + String.format("%.2f",s/360) +" Hours " );

}

public static void hourTo()
{
System.out.println();
System.out.println(h+ " Hours is equivalent to " + h*60 + "\tMinutes ");
System.out.println(h+ " Hours is equivalent to " + h*360 + "\tSeconds " );
}
} // End of class TimeConv

public class TimeConv
{
public static void main(String args[])
{
MyTime  D=new MyTime();
Scanner in=new Scanner(System.in);
System.out.println("My Time Conversion Program");
System.out.println("Time Conversion (Minutes and Seconds to Hours)");
System.out.println("~~~~ ~~~~~~~~~~ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");
System.out.print("Enter the minutes :  ");
D.m=in.nextInt();
System.out.print("Enter the seconds :  ");
D.s=in.nextInt();

D.toHour();
```

```
System.out.println("Time Conversion (Hours to Minutes and Seconds)");
System.out.println("~~~~ ~~~~~~~~~~~ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");
System.out.print("Enter the Hours :  ");
D.h=in.nextInt();
D.hourTo();

}

} //End of class TimeConv
```

OUTPUT

```
D:\Amudha\OOP JAVA 2023\UCS1312 LAB\JAVA PROGRAMS>javac TimeConv.java

D:\Amudha\OOP JAVA 2023\UCS1312 LAB\JAVA PROGRAMS>java TimeConv
My Time Conversion Program
Time Conversion (Minutes and Seconds to Hours)
~~~~ ~~~~~~~~~~ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Enter the minutes :  360
Enter the seconds :  7200

360.0 Minutes is equivalent to  6.00 Hours
7200.0 Seconds is equivalent to  20.00 Hours
Time Conversion (Hours to Minutes and Seconds)
~~~~ ~~~~~~~~~~ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Enter the Hours :  2

2.0 Hours is equivalent to 120.0      Minutes
2.0 Hours is equivalent to 720.0      Seconds
```

RESULT

Thus, the Java program to perform conversions between times components like hour, minutes and seconds was executed and verified.

EX.NO.3 JAVA PROGRAM TO IMPLEMENT MATRIX MULTIPLICATION USING 2D ARRAYS

AIM

To implement a Java program to perform multiplication of 2 two-dimensional integer arrays.

ALGORITHM

1. Define a class **MatMul** to perform multiplication of 2 matrices
2. Create two 3 X 3 integer matrices A[3,3] and B[3,3]
3. Initialize A and B with numeric values
4. Create matrix C to store the product of two matrices.
5. Multiply each row of the first matrix with every column of the second matrix and then add the results.
   Formula C[i][j] + = A[i][k] * B[k][j];
6. Print input matrices A & B
7. Print the product matrix C.

Matrix A

A11	A12	A13
A21	A22	A23
A31	A32	A33

Matrix B

B11	B12	B13
B21	B22	B23
B31	B32	B33

Matrix C

Row 1
C11 = (A11 * B11) + (A12 * B21) + (A13 * B31)
C12 = (A11 * B12) + (A12 * B22) + (A13 * B32)
C12 = (A11 * B13) + (A12 * B23) + (A13 * B33)
Row 2
C21 = (A21 * B11) + (A22 * B21) + (A23 * B31)
C22 = (A21 * B12) + (A22 * B22) + (A23 * B32)
C22 = (A21 * B13) + (A22 * B23) + (A23 * B33)
Row 3
C31 = (A31 * B11) + (A32 * B21) + (A33 * B31)
C32 = (A31 * B12) + (A32 * B22) + (A33 * B32)
C32 = (A31 * B13) + (A32 * B23) + (A33 * B33)

PROGRAM

File Name : MatMul.java

```
import java.io.*;

public class MatMul
{
public static void main(String args[])
{
int a[][]={{1,1,1},{2,2,2},{3,3,3}};
int b[][]={{1,1,1},{2,2,2},{3,3,3}};

int c[][]=new int[3][3];

System.out.println("Matrix A");
System.out.println("--------");

for(int i=0;i<3;i++)
{
for(int j=0;j<3;j++)
{
System.out.print(a[i][j]+"\t");
}
System.out.println("");
}
System.out.println();

System.out.println("Matrix B");
System.out.println("--------");

for(int i=0;i<3;i++)
{
for(int j=0;j<3;j++)
{
System.out.print(b[i][j]+"\t");
}
System.out.println("");
}
System.out.println();
//PRODUCT OF 2 MATRICES
System.out.println("Matrix C [PRODUCT OF 2 MATRICES]");
System.out.println("--------------------------------");



for(int i=0;i<3;i++)
{
for(int j=0;j<3;j++)
{
c[i][j]=0;
for(int k=0;k<3;k++)
{
    c[i][j]+=a[i][k]*b[k][j];
}
System.out.print(c[i][j]+"\t");
}
System.out.println();
}
}  //End of main
}  //End of class
```

OUTPUT

```
Matrix A
--------
1      1      1
2      2      2
3      3      3

Matrix B
--------
1      1      1
2      2      2
3      3      3

Matrix C [PRODUCT OF 2 MATRICES]
--------------------------------
6      6      6
12     12     12
18     18     18
```

RESULT

Thus the java program to implement 2 D multiplication on integers was written and tested.

EX.NO. 4 JAVA PROGRAM TO IMPLEMENT STRING HANDLING METHODS

AIM

To write a java program to implement string handling methods.

ALGORITHM

1. Import the necessary packages java.io.* and java.util.*;
2. Create a class StringManip to implement string handling methods.
3. Declare an ArrayList object of stype string to store a set of strings.
4. Design a menu driven code to implement the following operations

i) APPEND : list1.add(str)

ii) INSERT : list1.add(index,str)

iii) SEARCH : list1.indexOf(str)

iv) FIND STRINGS WITH STARTING LETTER : str.startsWith©

v) SIZE : list1.size()

vi) REMOVE : list1.remove(str)

vii) SORT: Collections.sort(list1)

5. Get user choice and invoke the respective method

PROGRAM
File Name: StringManip.java

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

public class StringManip
{
public static void main(String args[]) throws IOException
{
ArrayList<String> list1 = 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 starts 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 a string ");
str=in.readLine();
list1.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());
list1.add(i-1,str);
System.out.println("The array list has following elements:"+list1);
break;
case 3:
System.out.println("Enter the string to search ");
str=in.readLine();
j=list1.indexOf(str);
if(j==-1)
  System.out.println("Element not found");
else
  System.out.println("Index of "+str+"is "+(j+1));
break;

case 4:
System.out.println("Enter the character to List string that starts with specified character");
str=in.readLine();
for(i=0;i<(list1.size()-1);i++)
{
str1=list1.get(i);
if(str1.startsWith(str))
   System.out.println(str1);
}
break;

case 5:
System.out.println("Size of the list "+list1.size());
break;

case 6:
System.out.println("Enter the string to be removed");
str=in.readLine();

if(list1.remove(str))
        System.out.println("Element Removed"+str);
else
        System.out.println("Element not present");
break;

case 7:
Collections.sort(list1);
System.out.println("The array list has following elements:"+list1);
break;

case 8:
System.out.println("The array list has following elements:"+list1);
break;
}
```

```
System.out.println("Please Enter 0 to break and 1 to continue");
ch=Integer.parseInt(in.readLine());

}while(ch==1);
} //End of main
} //End of class
```

OUTPUT

```
D:\E DRIVE\OOP Java\LAB\UCS1312 LAB\JAVA PROGRAMS>javac StringManip.java
D:\E DRIVE\OOP Java\LAB\UCS1312 LAB\JAVA PROGRAMS>java StringManip
STRING MANIPULATION
************************
1. Append at end        2.Insert at particular index    3.Search
4. List string that starts with letter
5. Size        6.Remove        7.Sort        8.Display
Enter the choice
1
Enter a string
Apple
Please 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 that starts with letter
5. Size        6.Remove        7.Sort        8.Display
Enter the choice
1
Enter a string
Banana
Please 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 that starts with letter
5. Size        6.Remove        7.Sort        8.Display
Enter the choice
1
Enter a string
Carrot
Please 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 that starts with letter
5. Size        6.Remove        7.Sort        8.Display
Enter the choice
1
Enter a string
Egg
Please 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 that starts with letter
5. Size        6.Remove      7.Sort        8.Display
Enter the choice
1
Enter a string
Grapes
Please 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 that starts with letter
5. Size        6.Remove      7.Sort        8.Display
Enter the choice
2
Enter the string
Honey
Specify the index/position to insert
2
The array list has following elements:[Apple, Honey, Banana, Carrot, Egg, Grapes]
Please 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 that starts with letter
5. Size        6.Remove      7.Sort        8.Display
Enter the choice
3
Enter the string to search
Carrot
Index of Carrot is 4
Please 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 that starts with letter
5. Size        6.Remove      7.Sort        8.Display
Enter the choice
4
Enter the character to List string that starts with specified character
E
Egg
Please 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 that starts with letter
5. Size        6.Remove      7.Sort        8.Display
```

```
Enter the choice
5
Size of the list 6
Please 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 that starts with letter
5. Size      6.Remove      7.Sort       8.Display
Enter the choice
6
Enter the string to be removed
Banana
Element RemovedBanana
Please 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 that starts with letter
5. Size      6.Remove      7.Sort       8.Display
Enter the choice
8
The array list has following elements:[Apple, Honey, Carrot, Egg, Grapes]
Please 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 that starts with letter
5. Size      6.Remove      7.Sort       8.Display
Enter the choice
7
The array list has following elements:[Apple, Carrot, Egg, Grapes, Honey]
Please Enter 0 to break and 1 to continue
0
```

RESULT:

Thus the java program to implement string manipulation functionss are executed.

EX.NO. 5 JAVA PROGRAM TO IMPLEMENT CONSTRUCTORS

AIM

To write a java program that describes the constructor, overload the constructor and initiate its object.

ALGORITHM

1. Create a class as Box with data members width, height and depth.
2. Define default and parameterized constructors to initialize the members.
3. The first constructor Box ()is defined with no arguments. All members are initialized as 0.
4. The second constructor is defined with same name Box(double) and one argument to initialize width, height and depth with the same value.
5. The third constructor is created with same name Box(double,double,double) and 3 arguments to assign different values to the 3 members.
6. Define the method volume() to compute the volume of the box
7. Create different objects using the 3 different constructors.
8. Invoke volume() method using the 3 objects and print the volume.

PROGRAM
File Name : BoxConstructor.java

```
class Box
{
double width, height, depth;

Box(double w, double h, double d)
{
 width=w; height=h; depth=d;
}
Box()
{
width=height=depth=0;
}

Box(double length)
{
width=height=depth=length;
}

double volume()
{
return width*height*depth;
}

}

public class BoxConstructor
{

public static void main(String args[])
{
Box b1=new Box();
Box b2=new Box(10,20,15);
Box b3=new Box(7);
double vol;
vol=b1.volume();
System.out.println("Volume of box b1 is "+vol);
System.out.println("Volume of box b2 is "+b2.volume());
System.out.println("Volume of box b3 is "+b3.volume());
}
}
```

OUTPUT

```
D:\E DRIVE\OOP Java\LAB\UCS1312 LAB\JAVA PROGRAMS>javac BoxConstructor.java

D:\E DRIVE\OOP Java\LAB\UCS1312 LAB\JAVA PROGRAMS>java BoxConstructor
Volume of box b1 is 0.0
Volume of box b2 is 3000.0
Volume of box b3 is 343.0
```

RESULT

Thus the java program to implement constructors and constructor overloading was implemented.

EX.NO. 6 JAVA PROGRAM TO MANIPULATE CAR PRICE USING CLASS AND OBJECTS

AIM

To write a java program to compute the price of a car with tax by passing arguments to methods.

ALGORITHM

1. Create the class called car with the variables price, car name and tax rate with constantvalue.
2. Create the member function called total price to compute the price including tax.
3. The tax value is calculated by the actual rate and the tax rate as 12.5.
4. The total price is computed with the actual rate of the car and total amount of tax calculated for the corresponding car.
5. The first object c1 is created for the Car class and also the values are passed asargument for first model.
6. The second object c2 is created for the Car class and also the values are passed asargument for the second model.
7. Total value will be displayed for each Car model.

PROGRAM

File Name :

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

class Car
{
double price, tax, totalprice;
String carname;
double taxrate=12.5;

void Totalprice(double price,String carname)
{
tax=(price/100)*12.5;
totalprice=price+tax;
System.out.println("The total price of car " +carname +" is "+totalprice);
}

public static void main(String args[])
{
 Car c1=new Car();
 Scanner in=new Scanner(System.in);
 String model1,model2;
 double cost1,cost2;
 System.out.println("CAR 1");
 model1="Nissan";
 System.out.println("Model :\t "+model1);

 System.out.print("Cost :\t ");
 cost1=in.nextDouble();
 // c1.Totalprice(700000,"Nissan");
 c1.Totalprice(cost1,model1);

 Car c2 =new Car();
 System.out.println("CAR 2");
model2="Hyundai Creta";
System.out.println("Model :\t "+model2);

 System.out.print("Cost :\t ");
 cost2=in.nextDouble();
 c2.Totalprice(cost2,model2);
 //c2.Totalprice(110000,"Hyundai Creta");
}
}
```

OUTPUT

```
D:\E DRIVE\OOP Java\LAB\UCS1312 LAB\JAVA PROGRAMS>javac Car.java

D:\E DRIVE\OOP Java\LAB\UCS1312 LAB\JAVA PROGRAMS>java Car
CAR 1
Model :  Nissan
Cost :   700000
The total price of car Nissan is 787500.0
CAR 2
Model :  Hyundai Creta
Cost :   1100000
The total price of car Hyundai Creta is 1237500.0
```

RESULT

Thus the java program for tracking the price of a car is implemented and executed successfully.

EX.NO. 7 JAVA PROGRAM TO IMPLEMENT A BOOK CLASS

AIM

To write a java program to implement a Book class to manipulate array of objects.

ALGORITHM

1. Create the class Book with the variable bname, isbn, author and publisher.
2. Create the constructor for the class Book to initialize the variable.
3. Define constructors to initialize the values for bname, isbn, author and publisher.
4. Define the function getBookName(), getISBN(),getAuthor(), getPublisher() to fetch the book details.
5. Define the function displayInfo() and print the book details
6. Use this pointer int the member functions to access the members.
7. Create array of book objects and test the class functionalities.

PROGRAM

```
import java.io.*;

class Book
{
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
String bname;
int isbn;
String author;
String publisher;

Book()
{
bname="Java 2 : The Complete Reference";
isbn=1234567;
author="Herbert Schildt";
publisher="Tata Mc Graw Hills";
}

Book(String s1,int n,String s2,String s3)
{
this.bname=s1;
this.isbn=n;
this.author=s2;
this.publisher=s3;
}

public void displayInfo()
{
System.out.println("\n Book name "+bname);
System.out.println("\n ISBN number "+isbn);
System.out.println("\n Author name "+author);
System.out.println("\n Publisher name "+publisher);
}
}

public class TestBook
{
public static void main(String [] args) throws IOException
{
String book_name,book_author,book_publisher;
int book_isbn;
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
Book [] book = new Book [30];
int i=0;
System.out.print(" How many books info you want to store : ");
int n = Integer.parseInt(br.readLine());
book[0]=new Book();
System.out.println("BOOK 1 CREATED : Default - Constructor");
System.out.println("**************************************");
book[0].displayInfo();
System.out.println("***************************************");
for(i=1;i<n;i++)
{
System.out.printf("\n Enter book %d Details\n",i+1);
System.out.print("Enter book name : ");
book_name = br.readLine();
System.out.print("Enter book isbn : ");
book_isbn = Integer.parseInt(br.readLine());
System.out.print("Enter book author: ");
book_author = br.readLine();
System.out.print("Enter book publisher: ");
book_publisher= br.readLine();

book[i]=new Book(book_name,book_isbn,book_author,book_publisher);
}

System.out.println("***************************************");
System.out.println("************  PRINT BOOKS ************");
System.out.println("***************************************");
for(i=0;i<n;i++)
{
book[i].displayInfo();
System.out.println("***************************************");
}
}
}
```

OUTPUT

```
D:\E DRIVE\OOP Java\LAB\UCS1312 LAB\JAVA PROGRAMS>javac TestBook.java

D:\E DRIVE\OOP Java\LAB\UCS1312 LAB\JAVA PROGRAMS>java TestBook

How many books info you want to store : 3


BOOK 1 CREATED : Default - Constructor
**************************************

 Book name Java 2 : The Complete Reference

 ISBN number 1234567

 Author name Herbert Schildt

 Publisher name Tata Mc Graw Hills
**************************************

 Enter book 2 Details
Enter book name : Object Oriented Programming
Enter book isbn : 7654321
Enter book author: Balagurusamy
Enter book publisher: Pearson

Enter book 3 Details
Enter book name : Data Structures
Enter book isbn : 11112222
Enter book author: Mark Allen Weiss
Enter book publisher: Tata Mc Graw Hill


**************************************
*********** PRINT BOOKS ************
**************************************

 Book name Java 2 : The Complete Reference

 ISBN number 1234567

 Author name Herbert Schildt

 Publisher name Tata Mc Graw Hills
**************************************
```

```
Book name Object Oriented Programming

ISBN number 7654321

Author name Balagurusamy

Publisher name Pearson
****************************************

Book name Data Structures

ISBN number 11112222

Author name Mark Allen Weiss

Publisher name Tata Mc Graw Hill
****************************************
```

RESULT

Thus the java program to implement a Book class to manipulate array of objects was implemented.

EX.NO. 8 JAVA PROGRAM TO IMPLEMENT DATE CLASS WITH EXCEPTION HANDLING

AIM

To write a java program to implement a Date class to print date in specific format and raise exceptions in case of wrong input.

ALGORITHM

1. Import the necessary packages.
2. Define the user defined exception class InvalidDayException by extending the Exception class.
3. Define a constructor InvalidDayException() to pass message to Exception class using super keyword
4. Define the user defined exception class InvalidMonthException by extending the Exception class.
5. Define a constructor InvalidMonthException() to pass message to Exception class using super keyword
6. Define the class DateCheck with 3 data members day, month and year to store date, month and year.
7. Define a constructor DateCheck(){ } to initialize the members
8. Define member function dateFormat1() to print date in integer format dd/mm/yyyy [11/09/2023]
9. Define member function dateFormat2() to print date in string format dd-month-yyyy [17-September-2023]
10. Define the main function and read date input from the user.
11. Validate the day and raise the exception using throw statement.

    throw new InvalidDayException("MESSAGE");

12. Validate the month and raise the exception using throw statement.

    throw new InvalidMonthException("MESSAGE")

13. If no exceptions are raised, print the date in formats dd/mm/yyyy and dd/mm(string)/yyyy, by invoking the respective methods.
14. Compile and execute the program.

PROGRAM
File Name : DateCheck.java

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

class InvalidDayException extends Exception
{
 public InvalidDayException(String message)
 {
  super(message);
 }
}

class InvalidMonthException extends Exception
{
 public InvalidMonthException(String message)
 {
  super(message);
 }
}

class DateCheck
{
public static int day;
public static int month;
public static int year;
int flag=1;
DateCheck()
{
day=1;
month=1;
year=2023;
}


public static void dateFormat1(int d, int m, int y)
{
System.out.println("The given date is (dd/mm/yyyy) " +d + "/"+m+"/"+y);
}


public static void dateFormat2(int d,String m,int y)
{
System.out.println("The given date is (dd-MONTH-yyyy) " +d + "-"+m+"-"+y);
}
public static void main(String args[])throws Exception
{
Scanner in=new Scanner(System.in);
int flg=1;
String array[]={ "", "January", "February", "March", "April", "May", "June", "July", "August", "September",
"October", "November", "December" };
System.out.print("Enter the DATE :  ");
day=in.nextInt();
System.out.print("Enter the MONTH:  ");
month=in.nextInt();
System.out.print("Enter the YEAR :  ");
year=in.nextInt();

if(day<1 || day>31)
{
flg=0;
throw new InvalidDayException("Your day is invalid ! ");
}
if(month<1 || month>12)
{
flg=0;
throw new InvalidMonthException("Your month is invalid ! ");
}
if(flg==1)
{
System.out.print("Date in integral format : ");
dateFormat1(day,month,year);
System.out.print("Date in String format : ");
dateFormat2(day,array[month],year);
}

}

}
```

OUTPUT

```
D:\JAVA PROGRAMS>javac DateCheck.java

Run#1

D:\JAVA PROGRAMS>java DateCheck
Enter the DATE :  34
Enter the MONTH:  3
Enter the YEAR :  1987
Exception in thread "main" InvalidDayException: Your day is invalid !
      at DateCheck.main(DateCheck.java:57)

Run#2

D:\JAVA PROGRAMS>java DateCheck
Enter the DATE :  12
Enter the MONTH:  15
Enter the YEAR :  1987
Exception in thread "main" InvalidMonthException: Your month is invalid !
      at DateCheck.main(DateCheck.java:62)

Run#3

D:\JAVA PROGRAMS>java DateCheck
Enter the DATE :  17
Enter the MONTH:  9
Enter the YEAR :  2023
Date in integral format : The given date is (dd/mm/yy) 17/9/2023
Date in String format : The given date is (dd-MONTH-yy) 17-September-2023
```

RESULT

Thus, the java program to implement a user defined exception classes to validate date input and print date in specific format was executed.

EX.NO. 9 JAVA PROGRAM TO APPLY FILE HANDLING METHODS

AIM

To write a java program to implement java file handling methods on a specific file.

ALGORITHM

1. Import the necessary packages util.Scanner and io.File.
2. Read the file name from the input stream.
3. Create a file object f for the file name.
4. Print the details about the file, using the java built-in methods of File class
5. Print the file name using the method, f.getName()
6. Print the file path using the method, f.getPath()
7. Print the absolute path name of the file using the method, f.getAbsolutePath()
8. Print the parent using the method, f.getParent()
9. Print the file length using the method, f.length()
10. Print whether it is file or directory using the method, f.isFile()
11. Print whether the file name is valid using the method, f.isFile()
12. Print whether the file is readable using the method, f.canRead()
13. Print whether the file is Writable using the file name using the method, f.canWrite()
14. Print the file name using the method, f.isAbsolute()
15. Print the last modified details using the method f.lastModified()
16. To print in date and time format: Create object of SimpleDateFormat class

PROGRAM
File Name : FileDemo.java

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

class FileDemo
{
public static void main(String args[])
{
String fname;
Scanner s=new Scanner(System.in);
System.out.print("Enter the file name : ");
fname=s.nextLine();
File f1=new File(fname);
System.out.println("***************************");
System.out.println("\t\tFILE INFORMATION");
System.out.println("***************************");
System.out.println("NAME OF THE FILE :\t"+f1.getName());
System.out.println("PATH OF THE FILE :\t"+f1.getAbsolutePath());
System.out.println("PARENT  : \t\t"+f1.getParent());
if(f1.exists())
System.out.println("THE FILE EXISTS ");
else
System.out.println("THE FILE DOES NOT EXIST ");
if(f1.canRead())
System.out.println("THE FILE CAN BE READ ");
else
System.out.println("THE FILE CANNOT BE READ ");
if(f1.canWrite())
System.out.println("WRITE OPERATION IS PERMITTED");
else
System.out.println("WRITE OPERATION IS NOT PERMITTED");
if(f1.isDirectory())
System.out.println("IT IS A DIRECTORY ");
else
System.out.println("NOT A DIRECTORY");
if(f1.isFile())
System.out.println("IT IS A FILE ");
else
System.out.println("NOT A FILE");

SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");
System.out.println("File last modified on  : \t "+ sdf.format(f1.lastModified()));

System.out.println("LENGTH OF THE FILE : \t"+f1.length() + " bytes " );
System.out.println("FILE DELETED ! "+f1.delete());
}
}
```

INPUT FILE

File Name : *Story.txt*
Story Name : The Golden Egg
A farmer had a goose that laid one golden egg a day. He would sell the golden eggs, and they enjoyed a comfortable life. However, the farmer became greedy and wanted more than one egg a day. His wife foolishly agreed to his idea. The next day the farmer cut open the goose after it laid the golden egg. He could only find blood and guts. He realised his mistake. He now had no source of income, and the couple became poorer every day.

Moral: Think before you act.

OUTPUT

```
D:\ UCS1312 LAB\JAVA PROGRAMS>javac FileDemo.java

D:\ UCS1312 LAB\JAVA PROGRAMS>java FileDemo
Enter the file name : Story.txt
***************************
     FILE INFORMATION
***************************
NAME OF THE FILE :     Story.txt
PATH OF THE FILE :     D:\UCS1312 LAB\JAVA PROGRAMS\Story.txt
PARENT :          null
THE FILE EXISTS
THE FILE CAN BE READ
WRITE OPERATION IS PERMITTED
NOT A DIRECTORY
IT IS A FILE
File last modified on  :       05/09/2023 09:48:52
LENGTH OF THE FILE :   495 bytes
FILE DELETED ! true
```

RESULT

Thus, the java program to apply file handling methods was written and executed.

EX.NO. :10 JAVA PROGRAM TO STORE AND RETRIEVE STUDENT RECORD IN FILES

AIM

To write a java program to store and retrieve student record in files.

ALGORITHM

1. Import the necessary packages util.Scanner and io.File.

2 Define a method addRecord to read student details

3. Create an object for PrintWriter class with the file name.

PrintWriter pw = new PrintWriter(new BufferedWriter(new FileWriter("FileName.txt",true)));

4. Write the student details in file using the method pw.println()

5. Define readRecords() method to read file content line by line using FileReader object.

6. Print object details to console.

7. Define a method clear() to erase all file contents.

8. Define a method ShowMenu() to list all file operations to be done on the student data

9. Define a switch case menu driven process to invoke the respective method.

10. Define the main method and invoke showMenu() to get user choice.

11. Perform all file operations and print results.

PROGRAM
File Name : StudentData.java

```
import java.io.*;
class StudentData
{
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

public void addRecords() throws IOException
{

PrintWriter pw = new PrintWriter(new BufferedWriter(new FileWriter("Records.txt",true)));

String name, dept, fname, mname, address;
int age;
String s;
Long phoneno;
boolean addMore = false;

do
{
System.out.print("\nEnter name: ");
name = br.readLine();
System.out.print("Father's Name: ");
fname = br.readLine();
System.out.print("Mother's Name: ");
mname = br.readLine();
System.out.print("Address: ");
address = br.readLine();
System.out.print("Age: ");
age = Integer.parseInt(br.readLine());
System.out.print("\nEnter Department: ");
dept = br.readLine();
System.out.print("Telephone No.: ");
phoneno = Long.parseLong(br.readLine());
pw.println(name);
pw.println(fname);
pw.println(mname);
pw.println(address);
pw.println(age);
pw.println(dept);
pw.println(phoneno);

System.out.print("\nRecords added successfully !\n\nDo you want to add more records ? (y/n) : ");
s = br.readLine();
}
while(s.equalsIgnoreCase("y"));
pw.close();
showMenu();
}
```

```
public void readRecords() throws IOException
{
try
{
BufferedReader file = new BufferedReader(new FileReader("Records.txt"));
String name;
int i=1;
while((name = file.readLine()) != null)
{
System.out.println("*****************************");
System.out.println("Record.No. : " +(i++));
System.out.println("*****************************");
System.out.println("\nName: " +name);
System.out.println("Father's Name : "+file.readLine());
System.out.println("Mother's Name : "+file.readLine());
System.out.println("Address: "+file.readLine());
System.out.println("Age: "+Integer.parseInt(file.readLine()));
System.out.println("Dept: "+file.readLine());
System.out.println("Tel. No.: "+Long.parseLong(file.readLine()));
System.out.println();
}
file.close();
showMenu();
}
catch(FileNotFoundException e)
{
System.out.println("\nERROR : File not Found !!!");
}
}

public void clear() throws IOException
{
// Create a blank file
PrintWriter pw = new PrintWriter(new BufferedWriter(new FileWriter("Records.txt")));
pw.close();
System.out.println("\nAll Records cleared successfully !");
for(int i=0;i<999999999;i++); // Wait for some time
showMenu();
}

public void showMenu() throws IOException
{
System.out.print("1 : Add Records\n2 : Display Records\n");
System.out.print("3 : Clear All Records\n4 : Exit\n\nYour Choice : ");
int choice = Integer.parseInt(br.readLine());
switch(choice)
{
case 1:
addRecords();
break;
case 2:
readRecords();
break;
case 3:
clear();
break;
case 4:
System.exit(1);
break;
default:
System.out.println("\nInvalid Choice !");
break;
}
}
public static void main(String args[]) throws IOException
{
StudentData s = new StudentData();
s.clear();
s.showMenu();
}
}
```

OUTPUT

```
D:\JAVA PROGRAMS>javac StudentData.java
D:\JAVA PROGRAMS>java StudentData

1 : Add Records
2 : Display Records
3 : Clear All Records
4 : Exit

Your Choice : 1

Enter name: Bob
Father's Name: John
Mother's Name: Monika
Address: Coimbatore
Age: 18

Enter Department: CSE
Telephone No.: 1234567890

Records added successfully !

Do you want to add more records ? (y/n) : Y

Enter name: Pooja
Father's Name: Rakesh
Mother's Name: Nirmala
Address: New Delhi
Age: 17

Enter Department: CSE
Telephone No.: 9876543210

Records added successfully !

Do you want to add more records ? (y/n) : Y

Enter name: Manisha
Father's Name: Anil Kumar
Mother's Name: Geeta
Address: Haryana
Age: 18

Enter Department: ECE
Telephone No.: 6789054321

Records added successfully !
```

```
Do you want to add more records ? (y/n) : Y

Enter name: Jitender
Father's Name: Ganesh
Mother's Name: Sita
Address: Bangalore
Age: 17

Enter Department: Mech
Telephone No.: 1111222233

Records added successfully !

Do you want to add more records ? (y/n) : n
1 : Add Records
2 : Display Records
3 : Clear All Records
4 : Exit

Your Choice : 2

*****************************
Record.No. : 1
*****************************

Name: Bob
Father's Name : John
Mother's Name : Monika
Address: Coimbatore
Age:18
Dept: CSE
Tel. No.: 1234567890
*****************************
Record.No. : 2
*****************************

Name: Pooja
Father's Name : Rakesh
Mother's Name : Nirmala
Address: New Delhi
Age: 17
Dept: CSE
Tel. No.: 9876543210

*****************************
Record.No. : 3
*****************************

Name: Manisha
Father's Name : Anil Kumar
Mother's Name : Geeta
```

```
Address: Haryana
Age: 18
Dept: ECE
Tel. No.:  6789054321

****************************
Record.No. : 4
****************************

Name: Jitender
Father's Name : Ganesh
Mother's Name : Sita
Address: Bangalore
Age: 17
Dept: Mech
Tel. No.:  1111222233

1 : Add Records
2 : Display Records
3 : Clear All Records
4 : Exit

Your Choice : 3

All Records cleared successfully !

1 : Add Records
2 : Display Records
3 : Clear All Records
4 : Exit

Your Choice : 4
```

RESULT:

Thus, the java program to for student record maintenance in files was written and executed.

EX.NO.11 JAVA PROGRAM TO IMPLEMENT INTERFACE FOR CAR CLASS

AIM

To write a java program that uses a interface concepts to display the used car details.

ALGORITHM

1. Import the necessary packages util.Scanner and io.*
2. Define an interface secSalesItem with a method declaration getRetailPrice().
3. Define a class UsedCar and implement the interface secSalesItem
4. Define the methods getVehicleNumber(), getModel() and getPrice() in the class UsedCar.
5. Define the method getRetailPrice() to calculate the resale value of the car based on the amount, model and kilometers travelled.
6. Create an object c1 for UsedCar in the main method.
7. Invoke the methods getModel(), getOrice and getRetailPrice() and print the results.

PROGRAM

File Name : UsedCar.java

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


interface secSalesItem
{
void getRetailPrice();
}


class UsedCar implements secSalesItem
{
String vehicleNumber, model;
int year, kmTravelled;
public long price;
public static Scanner s=new Scanner(System.in);


public  void getVehicleNumber()
{
System.out.print("Type the vehicle number: ");
vehicleNumber=s.nextLine();
}


public  void getModel()
{
System.out.print("Type the car model : ");
model=s.nextLine();
System.out.print("Type the year of purchase : ");
year=s.nextInt();
System.out.print("Type the Kilometers Travelled : ");
kmTravelled=s.nextInt();
}
public  void getPrice()
{
System.out.print("Type the price of the car : ");
price=s.nextLong();
}
public void getRetailPrice()
{
int yrcr=2023-year;
int kmcr=0;
double resalecost;
int yearcredit[]={10,9,8,7,6,5,4,3,2,1};
if(kmTravelled>100000)
kmcr=0;
else if(kmTravelled>80000)
kmcr=1;
else if(kmTravelled>60000)
kmcr=2;
else if(kmTravelled>50000)
kmcr=3;
else if(kmTravelled>40000)
kmcr=4;
else if(kmTravelled<=40000)
kmcr=5;
int kmcredit[]={1,2,3,4,5,6,7,8,9,10};
resalecost=(price/100)*kmcredit*yearcredit[yrcr];
System.out.println("Resale value of your car is : Rs."+resalecost);
}


public static void main(String[] args)
{
System.out.println("*** USED CAR SALES ***");
UsedCar c1=new UsedCar();
c1.getModel();
c1.getPrice();
c1.getRetailPrice();
}

}
```

OUTPUT:

```
D:\JAVA PROGRAMS>javac UsedCar.java

D:\JAVA PROGRAMS>java UsedCar
*** USED CAR SALES ***
Type the car model : Santro
Type the year of purchase : 2020
Type the Kilometers Travelled : 60000
Type the price of the car : 300000
Resale value of your car is : Rs. 63000.0
```

RESULT:

Thus, the java program to calculate the resale value of used car using interface was implemented.

CONTENT BEYOND SYLLABUS

EX.NO. :12 JAVA PROGRAM TO IMPLEMENT MULTI THREADING

AIM

To write a java program to implement the concept of multiple threads in an application.

ALGORITHM

```
1. Import the necessary packages util.Scanner and io.*
2. Create a Class ThreadDemo implementing the Runnable interface
3. Define the constructor to initialize the thread name.
  ThreadDemo( String name)
  {
    threadName = name;
  }

4. Define the run() method with task to be executed for the thread.
5. Include necessary exception handling routines
6. Define the start() method to instantiate a thread object t, and execute the thread using
t.start().
7. Define main() method in the class TestThread
8. Create thread objects for ThreadDemo class.
9. Run the threads using the method start();
10. Print the results and analyze the output.
```

PROGRAM

```
import java.io.*;

class ThreadDemo implements Runnable
{

  private Thread t;
  private String threadName;

  ThreadDemo( String name)
  {
    threadName = name;
    System.out.println("Creating " +  threadName );
  }

  public void run()
  {

   System.out.println("Running " +  threadName );
    try
    {
      for(int i = 4; i > 0; i--)
      {
        System.out.println("Thread: " + threadName + ", " + i);
        Thread.sleep(50);
      }
    }
    catch (InterruptedException e)
    {
      System.out.println("Thread " +  threadName + " interrupted.");
     }
    System.out.println("Thread " +  threadName + " exiting.");
  }

  public void start ()
 {
    System.out.println("Starting " +  threadName );
    if (t == null) {
      t = new Thread (this, threadName);
      t.start ();
  }
 }
}
```

```
public class TestThread
{
  public static void main(String args[])
  {
    ThreadDemo R1 = new ThreadDemo( "Thread-1");
    R1.start();

    ThreadDemo R2 = new ThreadDemo( "Thread-2");
    R2.start();

    ThreadDemo R3 = new ThreadDemo( "Thread-3");
    R3.start();
  }
}
```

OUTPUT

```
D:\JAVA PROGRAMS>javac TestThread.java

D:\ JAVA PROGRAMS>java TestThread

Creating Thread-1
Starting Thread-1
Creating Thread-2
Starting Thread-2
Running Thread-1
Creating Thread-3
Running Thread-2
Starting Thread-3
Running Thread-3
Thread: Thread-3, 4
Thread: Thread-1, 4
Thread: Thread-2, 4
Thread: Thread-3, 3
Thread: Thread-1, 3
Thread: Thread-2, 3
Thread: Thread-3, 2
Thread: Thread-2, 2
Thread: Thread-1, 2
Thread: Thread-3, 1
Thread: Thread-2, 1
Thread: Thread-1, 1
Thread Thread-2 exiting.
Thread Thread-3 exiting.
Thread Thread-1 exiting.
```

RESULT:

Thus, the java program to learn the concept of multi-threading was implemented.

EX.NO. 13 JAVA PROGRAM TO IMPLEMENT FOR EVENT DRIVEN APPLICATION DEVELOPMENT

AIM

To write a java program to develop an event driven application to design a scientific calculator to support all arithmetic operations.

ALGORITHM

1. Import the required packages for I/O operations and swing operations.
2. Define a class Calc that extends JFrame and implements ActionListener.
3. Define a constructor to create a new frame with title "Calculator", buttons and textboxes.
4. Define the method actionPerformed() to detect the key pressed and update the textbox with the number or symbol that was pressed.
5. For every component include the Listeners using the java method addActionListener()
6. Use getSource() method to identify the button pressed and do necessary calculation and print the result in the text box.
7. Define the class SwingDecCalculator to create an object for the Calc class and invoke the frame creation.
8. Run the program
9. Give necessary decimal and integral values and test the results.

PROGRAM

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

class Calc extends JFrame implements ActionListener
{
  JFrame f;
  JTextField t;
  JButton b1,b2,b3,b4,b5,b6,b7,b8,b9,b0,bdiv,bmul,bsub,badd,bdec,beq,bdel,bclr;

  static double a=0,b=0,result=0;
  static int operator=0;

  Calc()
  {
    f=new JFrame("My Personal Calculator");
    t=new JTextField();
    b1=new JButton("1");
    b2=new JButton("2");
    b3=new JButton("3");
    b4=new JButton("4");
    b5=new JButton("5");
    b6=new JButton("6");
    b7=new JButton("7");
    b8=new JButton("8");
    b9=new JButton("9");
    b0=new JButton("0");
    bdiv=new JButton("/");
    bmul=new JButton("*");
    bsub=new JButton("-");
    badd=new JButton("+");
    bdec=new JButton(".");
    beq=new JButton("=");
    bdel=new JButton("BackSpace");
    bclr=new JButton("Clear");

    t.setBounds(30,40,280,30);
    b7.setBounds(40,100,50,40);
    b8.setBounds(110,100,50,40);
    b9.setBounds(180,100,50,40);
    bdiv.setBounds(250,100,50,40);
```

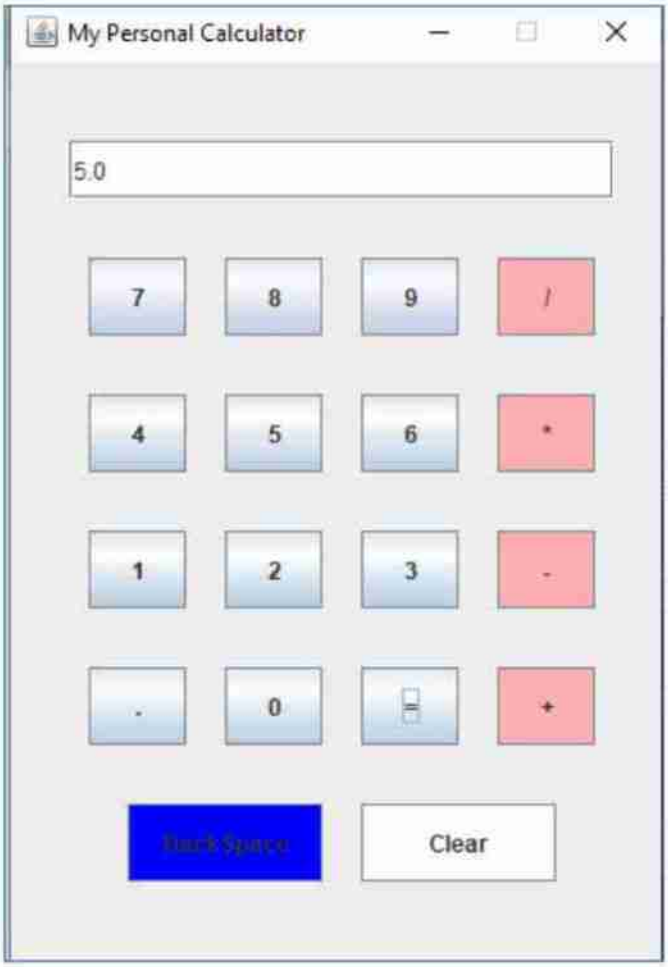
```
b4.setBounds(40,170,50,40);
b5.setBounds(110,170,50,40);
b6.setBounds(180,170,50,40);
bmul.setBounds(250,170,50,40);

b1.setBounds(40,240,50,40);
b2.setBounds(110,240,50,40);
b3.setBounds(180,240,50,40);
bsub.setBounds(250,240,50,40);

bdec.setBounds(40,310,50,40);
b0.setBounds(110,310,50,40);
beq.setBounds(180,310,50,40);
badd.setBounds(250,310,50,40);

bdel.setBounds(60,380,100,40);
bclr.setBounds(180,380,100,40);


badd.setBackground(Color.pink);
bsub.setBackground(Color.pink);
bmul.setBackground(Color.pink);
bdiv.setBackground(Color.pink);
bdel.setBackground(Color.blue);
bclr.setBackground(Color.white);
f.add(t);
f.add(b7);
f.add(b8);
f.add(b9);
f.add(bdiv);
f.add(b4);
f.add(b5);
f.add(b6);
f.add(bmul);
f.add(b1);
f.add(b2);
f.add(b3);
f.add(bsub);
f.add(bdec);
f.add(b0);
f.add(beq);
f.add(badd);
f.add(bdel);
f.add(bclr);

f.setLayout(null);
f.setVisible(true);
```

```
    f.setSize(350,500);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.setResizable(false);

    b1.addActionListener(this);
    b2.addActionListener(this);
    b3.addActionListener(this);
    b4.addActionListener(this);
    b5.addActionListener(this);
    b6.addActionListener(this);
    b7.addActionListener(this);
    b8.addActionListener(this);
    b9.addActionListener(this);
    b0.addActionListener(this);
    badd.addActionListener(this);
    bdiv.addActionListener(this);
    bmul.addActionListener(this);
    bsub.addActionListener(this);
    bdec.addActionListener(this);
    beq.addActionListener(this);
    bdel.addActionListener(this);
    bclr.addActionListener(this);
}

public void actionPerformed(ActionEvent e)
{
    if(e.getSource()==b1)
      t.setText(t.getText().concat("1"));
    if(e.getSource()==b2)
      t.setText(t.getText().concat("2"));

    if(e.getSource()==b3)
      t.setText(t.getText().concat("3"));
    if(e.getSource()==b4)
      t.setText(t.getText().concat("4"));

    if(e.getSource()==b5)
      t.setText(t.getText().concat("5"));
   if(e.getSource()==b6)
      t.setText(t.getText().concat("6"));

    if(e.getSource()==b7)
      t.setText(t.getText().concat("7"));
    if(e.getSource()==b8)
      t.setText(t.getText().concat("8"));
```

```
if(e.getSource()==b9)
   t.setText(t.getText().concat("9"));
if(e.getSource()==b0)
   t.setText(t.getText().concat("0"));

if(e.getSource()==bdec)
   t.setText(t.getText().concat("."));


if(e.getSource()==badd)
{
   a=Double.parseDouble(t.getText());
   operator=1;
   t.setText("");
}

if(e.getSource()==bsub)
{
   a=Double.parseDouble(t.getText());
   operator=2;
   t.setText("");
}

if(e.getSource()==bmul)
{
   a=Double.parseDouble(t.getText());
   operator=3;
   t.setText("");
}

if(e.getSource()==bdiv)
{
   a=Double.parseDouble(t.getText());
   operator=4;
   t.setText("");
}
```

```
      if(e.getSource()==beq)
      {
         b=Double.parseDouble(t.getText());

         switch(operator)
         {
            case 1: result=a+b;
               break;

            case 2: result=a-b;
               break;

            case 3: result=a*b;
               break;

            case 4: result=a/b;
               break;

            default: result=0;
         }

         t.setText(""+result);
      }

      if(e.getSource()==bclr)
         t.setText("");

      if(e.getSource()==bdel)
      {
         String s=t.getText();
         t.setText("");
         for(int i=0;i<s.length()-1;i++)
         t.setText(t.getText()+s.charAt(i));
      }
   }
}
class SwingCalculator
{
   public static void main(String[] args)
   {
      Calc frame = new Calc();
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      frame.setVisible(true);
   }
}
```

OUTPUT



5.0

RESULT

Thus a java based Calculator that supports both decimal and scientific calculations was designed using event driven programming.