* Implementing stack operations:
* using System;
* public class Stack
* {
* private int top = -1;
* private int size;
* private int[] stack;
* public Stack(int size ){
* this.size = size;
* stack = new int[size];
* }
* public bool IsUnderFlow(){
* if(top<0)
* return true;
* else
* return false;
* }
* public bool IsOverFlow(){
* if(top >size)
* return true;
* else
* return false;
* }
* public void Push(int s){
* if(IsOverFlow())
* Console.WriteLine("Stack is Full\n");
* stack[++top] = s;
* Console.WriteLine("Element Added\n");
* }
* public void Pop(){
* if(IsUnderFlow())
* Console.WriteLine("Stack is Empty\n");
* else
* {
* top--;
* Console.WriteLine("Element Removed\n");
* }
* }
* public void Disp(){
* if(IsUnderFlow())
* Console.WriteLine("Stack is Empty");
* else
* {
* Console.WriteLine("Elements Are:");
* for(int i = 0;i<=top;i++)
* {
* Console.WriteLine(stack[i]);
* }
* Console.Write("\n");
* }
* }
* public void Peek(){
* if(IsUnderFlow())
* Console.WriteLine("Stack is Empty\n");
* else
* {
* Console.WriteLine(stack[top]);
* }
* }
* }
* class Test
* {
* public static void Main()
* {
* Stack s=new Stack(10);
* while(true)
* {
* Console.WriteLine("1 to Add element\n2 to Remove Element\n3 to Display elements\n4 to Peek\n5 to Exit");
* int choice=Convert.ToInt32(Console.ReadLine());
* switch(choice)
* {
* case 1:
* {
* Console.Write("Enter element:");
* int ele=Convert.ToInt32(Console.ReadLine());
* s.Push(ele);
* continue;
* }
* case 2:
* {
* s.Pop();
* continue;
* }
* case 3:
* {
* s.Disp();
* continue;
* }
* case 4:
* {
* s.Peek();
* break;
* }
* case 5:
* {
* Environment.Exit(0);
* break;
* }
* }
* }
* }
* }

Output:





Operator Overloading:

using System;

class Complex

{

    private int real, img;

    public Complex(int r, int i)

    {

        real=r;

        img=i;

    }

    public static Complex operator +(Complex al, Complex a2)

    {

        Complex temp =new Complex (al.real + a2.real,al.img + a2.img);

        return temp;

    }

    public override string ToString()

    {

        return(string.Format("{0}+{1}i", real,img));

    }

}

class Test1

{

    public static void Main()

    {

        Complex al=new Complex(2,3);

        Complex a2=new Complex(4,5);

        Complex a3=al+a2;

        Console.WriteLine(a3);

    }

}

Output:

6+8i

Extracting string without using string method:

using System;

namespace Tejas{

class String

{

    char []str;

    int len;

    public String()

    {

        str=null;

        len=0;

    }

    public String(string s)

    {

        len=s.Length;

        str= new char[len];

        s.CopyTo(0,str,0,len);

    }

    public override string ToString()

    {

        return( new string(str));

    }

    public static void sub(String s, int start, int end)

    {

        String s2=new String();

        s2.str= new char[end];

        for(int i=start-1, j=0; i<end; i++, j++)

        {

            s2.str[j]=s.str[i];

        }

        Console.WriteLine(s2.str);

    }

}

}

class try1

{

    public static void Main()

    {

        Tejas.String s= new Tejas.String ("Tejas");

        Console.WriteLine("Enter starting index");

        int start=Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Enter ending index");

        int end=Convert.ToInt32(Console.ReadLine());

        Tejas.String.sub(s,start,end);

    }

}

Output:

Enter starting index

2

Enter ending index

3

ej

Sort list of strings with using userdefined function

using System;

class Program

{

    public static void Main()

    {

        Console.WriteLine("Enter the number of strings:");

        int n = int.Parse(Console.ReadLine());

        string[] strings = new string[n];

        Console.WriteLine("Enter the strings:");

        for (int i = 0; i < n; i++)

        {

            strings[i] = Console.ReadLine();

        }

        Sortstr(strings);

        Console.WriteLine("Sorted strings:");

        foreach (string str in strings)

        {

            Console.WriteLine(str);

        }

    }

    static void Sortstr(string[] arr)

    {

        int n = arr.Length;

        for (int i = 0; i < n - 1; i++)

        {

            for (int j = 0; j < n - i - 1; j++)

            {

                if (string.Compare(arr[j], arr[j + 1]) > 0)

                {

                    string temp = arr[j];

                    arr[j] = arr[j + 1];

                    arr[j + 1] = temp;

                }

            }

        }

    }

}

Output:

Enter the number of strings:

4

Enter the strings:

tejas

abhiram

varsha

ujwal

Sorted strings:

abhiram

tejas

ujwal

varsha

Program to implement arraylist methods

using System;

using System.Collections;

class Program

{

    public static void Main()

    {

        ArrayList arrayList = new ArrayList();

        // Add elements to the ArrayList

        arrayList.Add("Apple");

        arrayList.Add("Banana");

        arrayList.Add("Orange");

        // Print the elements

        Console.WriteLine("ArrayList elements:");

        PrintArrayList(arrayList);

        // Check if the ArrayList contains a specific element

        Console.WriteLine("Contains 'Banana': " + arrayList.Contains("Banana"));

        // Insert an element at a specific index

        arrayList.Insert(1, "Grapes");

        Console.WriteLine("ArrayList after inserting 'Grapes' at index 1:");

        PrintArrayList(arrayList);

        // Copy the elements to an array

        string[] array = new string[arrayList.Count];

        arrayList.CopyTo(array);

        // Remove an element from the ArrayList

        arrayList.Remove("Apple");

        Console.WriteLine("ArrayList after removing 'Apple':");

        PrintArrayList(arrayList);

        // Remove an element at a specific index

        arrayList.RemoveAt(1);

        Console.WriteLine("ArrayList after removing element at index 1:");

        PrintArrayList(arrayList);

        // Sort the ArrayList

        arrayList.Sort();

        Console.WriteLine("ArrayList after sorting:");

        PrintArrayList(arrayList);

        // Get the number of elements in the ArrayList

        Console.WriteLine("Number of elements in the ArrayList: " + arrayList.Count);

        // Clear the ArrayList

        arrayList.Clear();

        Console.WriteLine("ArrayList after clearing:");

        PrintArrayList(arrayList);

    }

    static void PrintArrayList(ArrayList arrayList)

    {

        foreach (var item in arrayList)

        {

            Console.WriteLine(item);

        }

        Console.WriteLine();

    }

}

Output:

ArrayList elements:

Apple

Banana

Orange

Contains 'Banana': True

ArrayList after inserting 'Grapes' at index 1:

Apple

Grapes

Banana

Orange

ArrayList after removing 'Apple':

Grapes

Banana

Orange

ArrayList after removing element at index 1:

Grapes

Orange

ArrayList after sorting:

Grapes

Orange

Number of elements in the ArrayList: 2

ArrayList after clearing:

Demonstrating the usage of Regular Expression:

using System;

using System.Text.RegularExpressions;

class Program

{

    // a regular expression pattern for a five-letter word

    // that starts with "a" and ends with "e"

    static string pattern = "^a...e$";

    public static void Main()

    {

        // create an instance of Regex class and

        //  pass the regular expression (i.e pattern)

        Regex rg = new Regex(pattern);

        Console.WriteLine("Enter string to match pattern of word 'starting with a and ending with e'");

        string str=Console.ReadLine();

        // IsMatch() returns true if "apple" matches the regular expression

        if (rg.IsMatch(str))

        {

            Console.WriteLine("String matches the pattern");

        }

        else

        {

            Console.WriteLine("String doesn't match the pattern");

        }

    }

}

Output:

Enter string to match pattern of word 'starting with a and ending with e'

apple

String matches the pattern

Program to implement user defined exception handling:

using System;

// Custom exception class

public class MyCustomException : Exception

{

    public MyCustomException(string message) : base(message)

    {

    }

}

// Main program

class Program

{

    public static void Main()

    {

        try

        {

            Console.WriteLine("Enter a number greater than 10:");

            int number = Convert.ToInt32(Console.ReadLine());

            if (number <= 10)

            {

                throw new MyCustomException("Number should be greater than 10.");

            }

            Console.WriteLine("You entered: " + number);

        }

        catch (MyCustomException ex)

        {

            Console.WriteLine("Custom Exception: " + ex.Message);

        }

        catch (FormatException)

        {

            Console.WriteLine("Invalid number format.");

        }

        catch (Exception ex)

        {

            Console.WriteLine("Exception: " + ex.Message);

        }

        Console.WriteLine("Program execution completed.");

    }

}

Output:

Enter a number greater than 10:

6

Custom Exception: Number should be greater than 10.

Program execution completed.

PS C:\c# programs\excephand> dotnet run \\ visual studio terminal

Enter a number greater than 10:

fesf

Invalid number format.

Program execution completed.

PS C:\c# programs\excephand> dotnet run \\ visual studio terminal

Enter a number greater than 10:

20

You entered: 20

Program execution completed.

Using Properties implement transaction in SB account.Deposit, Withdraw, interest calculation, balance enquiry and if balance falls below 1000rs for every 15 days impose penalty of 100rs.

using System;

class BankAccount

{

    private double balance;

    private DateTime lastTransactionDate;

    public double Balance

    {

        get { return balance; }

        private set { balance = value; }

    }

    public BankAccount(int x)

    {

        balance = x;

        lastTransactionDate = DateTime.Now;

    }

    public void Deposit(double amount)

    {

        balance += amount;

        lastTransactionDate = DateTime.Now;

       Console.WriteLine("Deposit of {0} Rs successful. New balance: {1} Rs", amount, balance);

    }

    public void Withdraw(double amount)

    {

        if (amount <= balance)

        {

            balance -= amount;

            lastTransactionDate = DateTime.Now;

            Console.WriteLine("Withdrawal of {0} Rs successful. New balance: {1} Rs", amount, balance);

        }

        else

        {

            Console.WriteLine("Insufficient balance");

        }

    }

    public void CalculateInterest(double interestRate)

    {

        double interest = balance \* interestRate / 100;

        balance += interest;

        Console.WriteLine("Interest calculated at {0}% rate. Interest amount: {1} Rs. New balance: {2} Rs", interestRate, interest, balance);

    }

    public void CheckBalance()

    {

        Console.WriteLine("Account balance: Rs " + balance);

    }

    public void ApplyPenalty()

    {

        DateTime currentDate = DateTime.Now;

        TimeSpan timeSinceLastTransaction = currentDate - lastTransactionDate;

        int daysSinceLastTransaction = timeSinceLastTransaction.Days;

        if (daysSinceLastTransaction >= 15 && balance < 1000)

        {

            balance -= 100;

           Console.WriteLine("Penalty of 100 Rs imposed. New balance: {0} Rs", balance);

        }

    }

}

class Program

{

    public static void Main()

    {

        BankAccount account = new BankAccount(1000);

        Console.WriteLine("Thank You For opening account in our bank\nYour Initial Balance is Rs 1000");

        while(true)

        {

           Console.WriteLine("Banking System Menu:");

            Console.WriteLine("1. Deposit");

            Console.WriteLine("2. Withdraw");

            Console.WriteLine("3. Calculate Interest");

            Console.WriteLine("4. Check Balance");

            Console.WriteLine("5. Exit");

            Console.Write("Enter your choice (1-5): ");

            int choice = Convert.ToInt32(Console.ReadLine());

            switch(choice)

            {

                case 1:

                {

                    Console.Write("Enter the amount to deposit: ");

                    double depositAmount = Convert.ToDouble(Console.ReadLine());

                    account.Deposit(depositAmount);

                    break;

                }

                case 2:

                {

                     Console.Write("Enter the amount to withdraw: ");

                    double withdrawAmount = Convert.ToDouble(Console.ReadLine());

                    account.Withdraw(withdrawAmount);

                    break;

                }

                case 3:

                {

                    account.CalculateInterest(5);

                    break;

                }

                case 4:

                {

                   account.CheckBalance();

                   break;

                }

                case 5:

                {

                    Environment.Exit(0);

                    break;

                }

                default:

                    Console.WriteLine("Invalid choice. Please try again.");

                    break;

            }

            account.ApplyPenalty();

        }

    }

}

Output:

Thank You For opening account in our bank

Your Initial Balance is Rs 1000

Banking System Menu:

1. Deposit

2. Withdraw

3. Calculate Interest

4. Check Balance

5. Exit

Enter your choice (1-5): 1

Enter the amount to deposit: 300

Deposit of 300 Rs successful. New balance: 1300 Rs

Banking System Menu:

1. Deposit

2. Withdraw

3. Calculate Interest

4. Check Balance

5. Exit

Enter your choice (1-5): 2

Enter the amount to withdraw: 5000

Insufficient balance

Banking System Menu:

1. Deposit

2. Withdraw

3. Calculate Interest

4. Check Balance

5. Exit

Enter your choice (1-5): 2

Enter the amount to withdraw: 100

Withdrawal of 100 Rs successful. New balance: 1200 Rs

Banking System Menu:

1. Deposit

2. Withdraw

3. Calculate Interest

4. Check Balance

5. Exit

Enter your choice (1-5): 3

Interest calculated at 5% rate. Interest amount: 60 Rs. New balance: 1260 Rs

Banking System Menu:

1. Deposit

2. Withdraw

3. Calculate Interest

4. Check Balance

5. Exit

Enter your choice (1-5): 4

Account balance: Rs 1260

Banking System Menu:

1. Deposit

2. Withdraw

3. Calculate Interest

4. Check Balance

5. Exit

Enter your choice (1-5): 5

To implement Indexers:

using System;

class MyCollection

{

    private int[] data;

    public MyCollection(int size)

    {

        data = new int[size];

    }

    public int this[int index]

    {

        get

        {

            if (index >= 0 && index < data.Length)

                return data[index];

            else

                return 0; // Return a default value if index is out of range

        }

        set

        {

            if (index >= 0 && index < data.Length)

                data[index] = value;

        }

    }

}

class Program

{

    public static void Main()

    {

        MyCollection collection = new MyCollection(5);

        collection[0] = 10;

        collection[1] = 20;

        collection[2] = 30;

        collection[3] = 40;

        collection[4] = 50;

        for (int i = 0; i < 5; i++)

        {

            Console.WriteLine("Element at index {0}: {1}", i, collection[i]);

        }

    }

}

Output:

Element at index 0: 10

Element at index 1: 20

Element at index 2: 30

Element at index 3: 40

Element at index 4: 50

Using multicast deligate deliver same message broadcast to console.output and file.

using System;

using System.IO;

class PrintToDevice

{

    public static void WriteToConsole(string s)

    {

        Console.WriteLine(s);

    }

    public delegate void PrintData(string s);

    public static void WriteToFile(string s)

    {

        FileStream fs=new FileStream("Test.txt",FileMode.Append, FileAccess.Write);

        StreamWriter sw=new StreamWriter(fs);

        sw.WriteLine(s);

        sw. Flush();

        sw.Close();

        fs.Close();

    }

}

 class Test

{

    public static void Main()

    {

        PrintToDevice.PrintData pd=new PrintToDevice. PrintData(PrintToDevice.WriteToConsole);

        pd+=new PrintToDevice. PrintData(PrintToDevice.WriteToFile);

        pd("Tejas");

    }

}

Output:

Tejas

// Note: A file will be created with “Test.txt” and Tejas will be printed in it

Implement Linked list using built in namespace

using System;

using System.Collections.Generic;

class Program

{

    public static void Main()

    {

        LinkedList<int> linkedList = new LinkedList<int>();

        Console.WriteLine("Enter a series of integers (0 to stop):");

        int num;

        while (true)

        {

            num = Convert.ToInt32(Console.ReadLine());

            if (num == 0)

                break;

            linkedList.AddLast(num);

        }

        Console.WriteLine("\nLinked List:");

        foreach (int value in linkedList)

        {

            Console.WriteLine(value);

        }

    }

}

Output:

Enter a series of integers (0 to stop):

24

56

73

0

Linked List:

24

56

73

Using multiple inheritance implement student result:

using System;

interface IResult

{

    void CalculateResult();

}

interface IInput

{

    void InputMarks();

}

class Student : IResult, IInput

{

    private string name;

    private string rollNo;

    private int[] marks = new int[5];

    private bool isFail;

    public string Name

    {

        get { return name; }

        set { name = value; }

    }

    public string RollNo

    {

        get { return rollNo; }

        set { rollNo = value; }

    }

    public bool IsFail

    {

        get { return isFail; }

    }

    public void InputMarks()

    {

        Console.WriteLine("Enter marks for subject 1:");

        marks[0] = GetValidMark();

        Console.WriteLine("Enter marks for subject 2:");

        marks[1] = GetValidMark();

        Console.WriteLine("Enter marks for subject 3:");

        marks[2] = GetValidMark();

        Console.WriteLine("Enter marks for subject 4:");

        marks[3] = GetValidMark();

        Console.WriteLine("Enter marks for subject 5:");

        marks[4] = GetValidMark();

    }

    public void CalculateResult()

    {

        isFail = false;

        int totalMarks = 0;

        float percent;

        foreach (int mark in marks)

        {

            if (mark < 35)

            {

                isFail = true;

            }

            totalMarks += mark;

        }

        percent=totalMarks/5;

        if (isFail)

        {

            Console.WriteLine("Sorry, {0} (Roll No: {1}). You have failed.", name, rollNo);

        }

        else

        {

            Console.WriteLine("Congratulations, {0} (Roll No: {1})! You have passed with {2} marks. You have got {3}%", name, rollNo, totalMarks,percent);

        }

    }

    private int GetValidMark()

    {

        tt:

        int mark = Convert.ToInt32(Console.ReadLine());

        if (mark < 0 || mark > 100)

        {

            Console.WriteLine("Invalid mark entered. Please enter a mark between 0 and 100.");

            goto tt;

        }

        return mark;

    }

}

class Program

{

    public static void Main()

    {

        Student student = new Student();

        Console.Write("Enter student name: ");

        student.Name = Console.ReadLine();

        Console.Write("Enter student roll number: ");

        student.RollNo = Console.ReadLine();

        int choice = 0;

        while (choice != 3)

        {

            Console.WriteLine("Student Result Menu:");

            Console.WriteLine("1. Input Marks");

            Console.WriteLine("2. Calculate Result");

            Console.WriteLine("3. Exit");

            Console.Write("Enter your choice (1-3): ");

            choice = Convert.ToInt32(Console.ReadLine());

            switch (choice)

            {

                case 1:

                    student.InputMarks();

                    break;

                case 2:

                    student.CalculateResult();

                    break;

                case 3:

                    break;

                default:

                    Console.WriteLine("Invalid choice. Please try again.");

                    break;

            }

            Console.WriteLine();

        }

        Console.WriteLine("Thank you for using the student result system. Have a nice day!");

    }

}

Output:

Enter student name: tejas

Enter student roll number: 111

Student Result Menu:

1. Input Marks

2. Calculate Result

3. Exit

Enter your choice (1-3): 1

Enter marks for subject 1:

35

Enter marks for subject 2:

56

Enter marks for subject 3:

56

Enter marks for subject 4:

78

Enter marks for subject 5:

76

Student Result Menu:

1. Input Marks

2. Calculate Result

3. Exit

Enter your choice (1-3): 2

Congratulations, tejas (Roll No: 111)! You have passed with 301 marks. You have got 60%

Student Result Menu:

1. Input Marks

2. Calculate Result

3. Exit

Enter your choice (1-3): 3

Thank you for using the student result system. Have a nice day!