C# Hands-on

Account datails:

//Implement code here

using System;

public class Account

{

private int id;

private string accountType;

private double balance;

public int Id

{

get{ return id;}

set{ id = value;}

}

public string AccountType

{

get{ return accountType;}

set{ accountType = value;}

}

public double Balance

{

get{ return balance;}

set{ balance = value;}

}

public Account(){}

public Account(int id, string accountType, double balance)

{

this.id = id;

this.accountType = accountType;

this.balance = balance;

}

public bool WithDraw(double amount)

{

if(balance>amount)

{

balance -= amount;

return true;

}

return false;

}

public string GetDetails()

{

return ("Account Id: " + id + "\nAccount Type: " + accountType + "\nBalance: "+balance);

}

}

public class Program

{

static void Main(string[] args)

{

Account ac = new Account();

Console.WriteLine("Enter account id");

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

Console.WriteLine("Enter account type");

string accountType = Console.ReadLine();

Console.WriteLine("Enter account balance");

double balance = double.Parse(Console.ReadLine());

Console.WriteLine("Enter amount to withdraw");

double amount = double.Parse(Console.ReadLine());

ac = new Account(id, accountType, balance);

Console.WriteLine(ac.GetDetails());

if(ac.WithDraw(amount))

{

Console.WriteLine("New Balance: " + ac.Balance);

}

}

}

Append text to a file:

using System;

using System.IO;

public class Program //DO NOT change the class name

{

//implement code here

static void Main(string[] args)

{

Console.WriteLine("Enter the Sentence");

string i = Console.ReadLine();

File.AppendAllText("sentences.txt", i + Environment.NewLine);

i = Console.ReadLine();

File.AppendAllText("sentences.txt", i);

string readText = File.ReadAllText("sentences.txt");

Console.WriteLine(readText);

}

}

Boolean result:

using System;

public class Program //DO NOT change the class name

{

//implement code here

static void Main(string[] args)

{

int x,y;

bool result = false;

Console.WriteLine("Enter the value for x");

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

Console.WriteLine("Enter the value for y");

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

if(x<y)

result = true;

Console.WriteLine("The result of whether " + x + " is less than " + y + " is " + result);

}

}

Calculator program:

using System;

public class Calculator

{ public int Addition(int a, int b)

{

return (a+b);

}

public int Subtraction(int a, int b)

{

return(a-b);

}

public int Multiplication(int a, int b)

{

return(a\*b);

}

public double Division(int a, int b, out double remainder)

{

remainder = a%b;

return(Convert.ToInt32(a/b));

}

}

public class Program

{

static void Main(string[] args)

{

Calculator calc = new Calculator();

char operation;

int a,b;

Console.WriteLine("Enter the operator");

operation = Console.ReadLine()[0];

Console.WriteLine("Enter the operands");

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

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

switch(operation)

{

case '+':

Console.WriteLine("Result of " + a + operation + b + " is " + calc.Addition(a,b));

break;

case '-':

Console.WriteLine("Result of " + a + operation + b + " is " + calc.Subtraction(a,b));

break;

case '\*':

Console.WriteLine("Result of " + a + operation + b + " is " + calc.Multiplication(a,b));

break;

case '/':

double remainder;

Console.WriteLine("Result of " + a + operation + b + " is " + calc.Division(a,b,out remainder));

Console.WriteLine("Remainder =" + remainder);

break;

default:

Console.WriteLine("Invalid Operator");

break;

}

}

}

Change a word:

using System;

public class Program //DO NOT change the class name

{

//implement code here

static void Main(string[] args)

{

Console.WriteLine("Enter a string");

string str = Console.ReadLine();

if(str.Contains("the"))

{

str = str.Replace("the","that");

Console.WriteLine(str);

} else

{

Console.WriteLine("Word 'the' not found");

}

}

}

Copy file content:

using System;

using System.IO;

public class Program //DO NOT change the class name

{

//implement code here

static void Main(string[] args)

{

string writeText = File.ReadAllText("sample1.txt");

File.WriteAllText("sample2.txt", writeText);

string readText = File.ReadAllText("sample1.txt");

Console.WriteLine("From sample1.txt");

Console.WriteLine(readText);

readText = File.ReadAllText("sample2.txt");

Console.WriteLine("From sample2.txt");

Console.WriteLine(readText);

}

}

Display product details:

using System;

public class Product

{

private int productId;

private string productName;

private double price;

public int ProductId

{

get{return productId;}

set{productId = value;}

}

public string ProductName

{

get{return productName;}

set{productName = value;}

}

public double Price

{

get{return price;}

set{

if(value > 0)

price = value;

else

price = 0;

}

}

public void Display()

{

Console.WriteLine("Product ID: " + productId);

Console.WriteLine("Product Name: " + productName);

Console.WriteLine("Product Price: " + price);

}

}

public class Program //DO NOT change the class name

{

//implement code here

static void Main(string[] args)

{

Product p = new Product();

Console.WriteLine("Enter product id");

p.ProductId = int.Parse(Console.ReadLine());

Console.WriteLine("Enter product name");

p.ProductName = Console.ReadLine();

Console.WriteLine("Enter product price");

p.Price = double.Parse(Console.ReadLine());

p.Display();

}

}

Display long number:

using System;

namespace ProgFundamentals2

{

public class Program //DO NOT change the class name

{

//implement code here

static void Main(string[] args)

{

long num1, num2;

Console.WriteLine("Enter large number 1:");

num1 = Convert.ToInt64(Console.ReadLine());

Console.WriteLine("Enter large number 2:");

num2= Convert.ToInt64(Console.ReadLine());

Console.WriteLine("The result is: " + (num1+num2));

}

}

}

Extract book code:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ExtractBookCode //Do not change the namespace name

{

public class Program //Do not change the class name

{

public static void Main(String[] arg) //Do not change the method signature

{

//Implement code here

string book\_code;

Console.WriteLine("Enter the book code of length 18");

book\_code = Console.ReadLine();

if(book\_code.Length == 18)

{

if(book\_code.Substring(0,3) == "101" || book\_code.Substring(0,3) == "102" || book\_code.Substring(0,3)=="103")

{

Console.WriteLine("Department Code : " + book\_code.Substring(0,3));

} else

{

Console.WriteLine(" Invalid Department Code");

}

int year = int.Parse(book\_code.Substring(3,4));

if(year>=1900 && year<=2020)

{

Console.WriteLine("Year of Publication : " + year);

} else

{

Console.WriteLine("Invalid Year");

}

long pages = Int64.Parse(book\_code.Substring(7,5));

if(pages >= 00001 && pages<=99999)

{

Console.WriteLine("Number of Pages : " + pages);

} else

{

Console.WriteLine("Invalid Page Numbers");

}

char ch = Convert.ToChar(book\_code.Substring(12,1));

if(Char.IsLetter(ch))

{

if(Char.IsNumber(Convert.ToChar(book\_code.Substring(13,1))) && Char.IsNumber(Convert.ToChar(book\_code.Substring(14,1))) && Char.IsNumber(Convert.ToChar(book\_code.Substring(15,1))) && Char.IsNumber(Convert.ToChar(book\_code.Substring(16,1)))&& Char.IsNumber(Convert.ToChar(book\_code.Substring(17,1))))

{

Console.WriteLine("Book ID : " + book\_code.Substring(12,6));

}

} else

{

Console.WriteLine("Invalid Book ID");

}

} else

{

Console.WriteLine("Invalid Book Code");

}

}

}

}

Find square and cube:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Methods2 //DO NOT Change namespace name

{

public class Program //DO NOT Change class 'Program' name

{

public static void Main(string[] args) //DO NOT Change 'Main' method Signature

{

//Implement your code here

Console.WriteLine("Enter a Number");

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

double square = FindSquare(x);

double cube = FindCube(x);

Console.WriteLine("Square of " + x + " is " + square);

Console.WriteLine("Cube of " + x + " is " + cube);

}

//Implement methods here. Keep the method 'public' and 'static'

public static double FindSquare(double num){

return num\*num;

}

public static double FindCube(double num){

return num\*num\*num;

}

}

}

Find the age of a person:

using System;

public class Person //DO NOT change the class name

{

//implement code here

private string firstName, lastName;

DateTime dob;

public string FirstName

{

get { return firstName; }

set { firstName = value; }

}

public string LastName

{

get { return lastName; }

set { lastName = value; }

}

public DateTime Dob

{

get { return dob; }

set { dob = value; }

}

public string Adult

{

get

{

if(GetAge(dob) < 18)

return "Child";

else

return "Adult";

}

}

public void DisplayDetails()

{

Console.WriteLine("First Name: " + firstName);

Console.WriteLine("Last Name: " + lastName);

Console.WriteLine("Age: " + GetAge(dob));

Console.WriteLine(Adult);

}

public int GetAge(DateTime dob)

{

DateTime Now = DateTime.Now;

int age = 0;

age = DateTime.Now.Year - dob.Year;

if (DateTime.Now.DayOfYear < dob.DayOfYear)

age = age - 1;

return age;

}

}

class Program

{

static void Main(string[] args)

{

Person person = new Person();

Console.WriteLine("Enter first name");

person.FirstName = Console.ReadLine();

Console.WriteLine("Enter last name");

person.LastName = Console.ReadLine();

Console.WriteLine("Enter date of birth in yyyy/mm/dd/ format");

person.Dob = Convert.ToDateTime(Console.ReadLine());

person.DisplayDetails();

}

}

Game inheritance:

using System;

public class Game

{

public string Name {get; set;}

public int MaxNumPlayers {get; set;}

public override string ToString()

{

return ("Maximum number of players for " + Name + " is " + MaxNumPlayers);

}

}

public class GameWithTimeLimit : Game

{

public int TimeLimit {get; set;}

public override string ToString()

{

Console.WriteLine(base.ToString());

return ("Time Limit for " + Name + " is " + TimeLimit + " minutes");

}

}

public class Program

{

static void Main(string[] args)

{

Game g = new Game();

GameWithTimeLimit tl = new GameWithTimeLimit();

int time\_limit, max\_player\_1, max\_player\_2;

string game\_1, game\_2;

Console.WriteLine("Enter a game");

game\_1 = Console.ReadLine();

Console.WriteLine("Enter the maximum number of players");

max\_player\_1 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter a game that has time limit");

game\_2 = Console.ReadLine();

Console.WriteLine("Enter the maximum number of players");

max\_player\_2 = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter the time limit in minutes");

time\_limit = Convert.ToInt32(Console.ReadLine());

g.Name = game\_1;

g.MaxNumPlayers = max\_player\_1;

Console.WriteLine(g.ToString());

tl.Name = game\_2;

tl.MaxNumPlayers = max\_player\_2;

tl.TimeLimit = time\_limit;

Console.WriteLine(tl.ToString());

}

}

Generate bill details:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ProgFundamentals2 //DO NOT change the namespace name

{

public class Program //DO NOT change the class name

{

public static void Main(string[] args) //DO NOT change the 'Main' method signature

{

//Implement the code here

int pizzas, puffs, pepsi;

Console.Write("Enter the number of pizzas bought : ");

pizzas = Convert.ToInt32(Console.ReadLine())\*200;

Console.Write("Enter the number of puffs bought : ");

puffs = Convert.ToInt32(Console.ReadLine())\*40;

Console.Write("Enter the number of pepsi bought : ");

pepsi = Convert.ToInt32(Console.ReadLine())\*120;

int total = pizzas + puffs + pepsi;

Console.WriteLine("Bill Details");

Console.WriteLine("Cost of Pizzas: " + (pizzas));

Console.WriteLine("Cost of Puffs: " + (puffs));

Console.WriteLine("Cost of Pepsis: " + (pepsi));

Console.WriteLine("GST 12%: " + GST(total));

Console.WriteLine("CESS 5%: " + CESS(total));

Console.WriteLine("Total Price: " + total);

}

public static double GST(int amt){

return ((amt\*12)/100);

}

public static double CESS(int amt) {

return ((amt\*5)/100);

}

}

}

Load validator:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LoadValidator //Do not change the namespace

{

public class Program //Do not change the class name

{

static void Main(string[] args) //Do not change the method signature

{

//Implement code here

int people, load;

Console.WriteLine("Enter the number of people");

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

Console.WriteLine("Enter the total load in the elevator");

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

if(Validator.ValidateElevatorLoad(load))

{

Console.WriteLine("Ready to Go");

} else

{

Console.WriteLine("Maximum load Exceeded");

}

}

}

public class Validator //Do not change the class name

{

public static bool ValidateElevatorLoad(int elevatorLoad) //Do not change the method signature

{

//Implement code here

if(elevatorLoad < 1400)

return true;

return false;

}

}

}

Max value of signed byte:

using System;

public class Program //DO NOT change the class name

{

//implement code here

static void Main(string[] args)

{

sbyte number = 125;

Console.WriteLine("Value of number: " + number);

number = sbyte.MaxValue;

Console.WriteLine("Largest value stored in a signed byte : " + number);

}

}

Multiplication table charts:

using System;

public class Program //DO NOT change the class name

{

//implement code here

static void Main(string[] args)

{

int num;

Console.WriteLine("Enter the number");

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

for(int i = 1; i <= num; i++)

{

for(int j = 1; j <= 10; j++)

Console.Write((i\*j) + " ");

Console.WriteLine();

}

}

}

Openable interface:

using System;

interface IOpenable

{

string OpenSesame();

}

class TreasureBox : IOpenable

{

public string OpenSesame()

{

return ("Congratulations , Here is your lucky win");

}

}

class Parachute : IOpenable

{

public string OpenSesame()

{

return ("Have a thrilling experience flying in air");

}

}

class Program

{

static void Main(string[] args)

{

Parachute p = new Parachute();

TreasureBox t = new TreasureBox();

Console.WriteLine("Enter the letter found in the paper");

char ch = Console.ReadLine()[0];

if(ch == 'T')

Console.WriteLine(t.OpenSesame());

else if(ch == 'P')

Console.WriteLine(p.OpenSesame());

}

}

Quiz competition report:

using System;

namespace JaggedArray //DO NOT Change the namespace name

{

public class Program //DO NOT Change the class name

{

public static void Main(string[] args) //DO NOT change the method signature

{

//Implement code here

// Get input from the user and construct a jagged array

int teams;

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

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

int[] attempt = new int[teams];

for(int i = 0; i < teams; i++)

{

Console.WriteLine("No.of attempts for team " + (i+1) + ":");

attempt[i] = Convert.ToInt32(Console.ReadLine());

}

int max\_attempt = attempt[0];

for(int i = 0; i <teams; i++)

{

if(attempt[i] > max\_attempt)

{

max\_attempt = attempt[i];

}

}

int[][] score\_board = new int[teams][];

for(int i = 0; i <teams; i++)

{

Console.WriteLine("Enter the score for team " + (i+1) +":");

score\_board[i] = new int[attempt[i]];

for(int j = 0; j < attempt[i]; j++)

{

score\_board[i][j] = Convert.ToInt32(Console.ReadLine());

}

}

Console.WriteLine(GetTotalScore(score\_board));

}

public static String GetTotalScore(int[][] score) //DO NOT change the method signature

{

//Implement code here

//Method to calculate total score for each team and return a string as specified in the sample output.

string result = "";

int sum;

for(int i = 0; i < score.Length; i++)

{

sum = 0;

for(int j = 0; j <score[i].Length; j++)

{

sum += score[i][j];

}

result += "Team " + (i+1) + " Total Score is " + sum + ". ";

}

return result;

}

}

}

Reverse a sentence:

using System;

public class Program //DO NOT change the class name

{

//implement code here

static void Main(string[] args)

{

string str,result="";

Console.WriteLine("Enter a string");

str = Console.ReadLine();

string[] strarr = str.Split();

for(int i = strarr.Length-1; i >= 0; i--)

{

result += strarr[i] + " ";

}

Console.WriteLine(result);

}

}

String concatenate:

using System;

public class Program //DO NOT change the class name

{

//implement code here

static void Main(string[] args)

{

Console.WriteLine("Enter first name");

string fullName = Console.ReadLine();

Console.WriteLine("Enter last name");

fullName += " " + Console.ReadLine();

Console.WriteLine("Full name: " + fullName);

}

}

User details:

using System;

using System.Text;

public class Program //DO NOT change the class name

{

//implement code here

static void Main(string[] args)

{

string name,city;

float age;

char gender;

long mobile,pin;

Console.WriteLine("Enter name");

name = Console.ReadLine();

Console.WriteLine("Enter age(completed years and months)");

age = float.Parse(Console.ReadLine());

Console.WriteLine("Enter gender('M' for Male and 'F' for Female)");

gender = Console.ReadLine()[0];

Console.WriteLine("Enter city");

city = Console.ReadLine();

Console.WriteLine("Enter mobile number");

mobile = Convert.ToInt64(Console.ReadLine());

Console.WriteLine("Enter pincode");

pin = Convert.ToInt64(Console.ReadLine());

Console.WriteLine("Name: "+name);

Console.WriteLine("Age: "+age);

Console.WriteLine("Gender: "+gender);

Console.WriteLine("City: "+city);

Console.WriteLine("Mobile: "+mobile);

Console.WriteLine("Pincode: "+pin);

}

}

Vechile released year:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Linq.Expressions;

using System.Text;

using System.Threading.Tasks;

namespace LinqApp1 //DO NOT CHANGE the namespace name

{

public class Program //DO NOT CHANGE the class name

{

/\*\* DO NOT CHANGE this 'List' declaration with initialized values \*\*/

public static List<Vehicle> vehicleList = new List<Vehicle>()

{

new Vehicle("HO345","CRV","Honda",2015),

new Vehicle("HY562","Creta","Hyundai",2017),

new Vehicle("RE198","Duster","Reanult",2014),

new Vehicle("MA623","Spacio","Suzuki",2014),

new Vehicle("TR498","Nexon","Tata",2015),

new Vehicle("TR981","Zest","Tata",2016),

new Vehicle("HO245","WRV","Honda",2018)

};

static void Main(string[] args) //DO NOT Change this 'Main' signature

{

//Implement your code here

int y1,y2;

Console.WriteLine("Enter from year :");

y1=Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter to year :");

y2=Convert.ToInt32(Console.ReadLine());

getVehicleName(y1,y2);

}

//Implement the method 'getVehicleName' here

public static void getVehicleName(int fromYear, int toYear)

{

IEnumerable<string> L=from v in vehicleList

where(v.ReleaseYear>=fromYear && v.ReleaseYear<=toYear)

select v.VehicleName;

}

/\*\* DO NOT CHANGE this ParameterExpression \*\*/

public static ParameterExpression variableExpr = Expression.Variable(typeof(IEnumerable<Vehicle>), "sampleVar");

public static Expression getMyExpression(int fromYear, int toYear)

{

Expression assignExpr = Expression.Assign(variableExpr, Expression.Constant(vehicleList.Where(v=>v.ReleaseYear>=fromYear && v.ReleaseYear<=toYear).Select(n=>n)));

return assignExpr;

}

}

}

Flight status:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DateEx2 //DO NOT CHANGE the namespace name

{

public class Program //DO NOT CHANGE the class name

{

/\*\*\* Dictionary values are hard-coded. Do NOT change \*\*\*\*/

static Dictionary<string, DateTime> flightSchedule = new Dictionary<string, DateTime>(){

{"ZW346", Convert.ToDateTime("13:54:10")},

{"AT489", Convert.ToDateTime("16:30:00")},

{"BR267", Convert.ToDateTime("21:15:30")}};

public static void Main(string[] args) //DO NOT CHANGE the 'Main' method signature

{

//Implement your code here

string flight;

Console.WriteLine("Enter the Flight Number");

flight=Console.ReadLine();

Console.WriteLine(flightStatus(flight));

}

public static string flightStatus(string flightNo) //DO NOT CHANGE the 'flightStatus' method signature

{

//Implement your code here

if(!flightSchedule.ContainsKey(flightNo))

{

return "Invalid Flight Number";

}

DateTime departureTime=flightSchedule[flightNo];

if(DateTime.UtcNow<departureTime)

{

TimeSpan ts=departureTime.Subtract(DateTime.UtcNow);

return "Time to flight" +" "+ts.ToString();

}

else

{

return "Flight Already Left";

}

}

}

}

Balls bowled:

using System;

using System.Collections.Generic;

namespace BallsBowled //DO NOT change the namespace name

{

public class Program //DO NOT change the class name

{

static void Main(string[] args)

{

//Implement code here

Console.WriteLine("Enter the number of overs");

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

PlayerBO a=new PlayerBO();

a.AddOversDetails(oversBowled);

}

}

public class PlayerBO //DO NOT change the class name

{

int b;

public List<int> PlayerList { get; set; } = new List<int>();

public void AddOversDetails(int oversBowled) //DO NOT change the method signature

{

//Implement code here

PlayerList.Add(oversBowled);

foreach(int i in PlayerList)

{

b= i;

}

Console.WriteLine("Balls Bowled :" +GetNoOfBallsBowled());

}

public int GetNoOfBallsBowled() //DO NOT change the method signature

{

//Implement code here

int result=b\*6;

return result;

}

}

}