**Assignment of EAD**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**



**Submitted to:**

Dr.Shuja-ur-Rehman Baig

**Submitted by:**

Saad Ishtiaq

Roll no: BSEF18A025

***(DO NOT COPY.*** *This property belongs to***Saad Ishtiaq.** Beware***!!! )***

**Punjab University College of Information Technology,**

**PUCIT (Old Campus),**

**LAHORE**

**EAD HOMEWORK\_2**

Information is taken from

Provided Reference Book and Sir Lecture Notes.

**(BSSE & BSIT FALL 2018)**

**Q1:** Write down the advantages of unary and binary operators.

**Ans:**

**Unary Operators:**

Unary operators are operators which act upon a single operand to produce a new value.

**Examples:**

//Post Fix

int a = 5;

int b = a++;

Console.WriteLine($"a = {a} b={b}");

/\*------output: a = 6 b=5-----------\*/

//Pre Fix

int c = 5;

int d = ++c;

Console.WriteLine($"a = {c} b={d}");

/\*------output: a = 6 b=6-----------\*/

//Negation

bool x = false;

Console.WriteLine(!x);

/\*------output: True-----------\*/

**Binary Operators:**

Binaryoperators are those operators which act upon a two operands to produce a new value.

**Examples:**

int e = 6;

int f = 4;

Console.WriteLine($" e + f = {e + f} ");

Console.WriteLine($" e - f = {e - f} ");

Console.WriteLine($" e \* f = {e \* f} ");

Console.WriteLine($" e / f = {e / f} ");

Console.WriteLine($" e % f = {e % f} ");

/\*------output:

e + f = 10

e - f = 2

e \* f = 24

e / f = 1

e % f = 2

-----------\*/

**Q1:** Suppose a,b,c,d,e,f,g are int type variables and initialized with 0 value. What will be the output of these variables if we apply the given operations. Please attach screenshot of the output and explain the output values procedure clearly.

b = 39;

a = ++b;

c = --b;

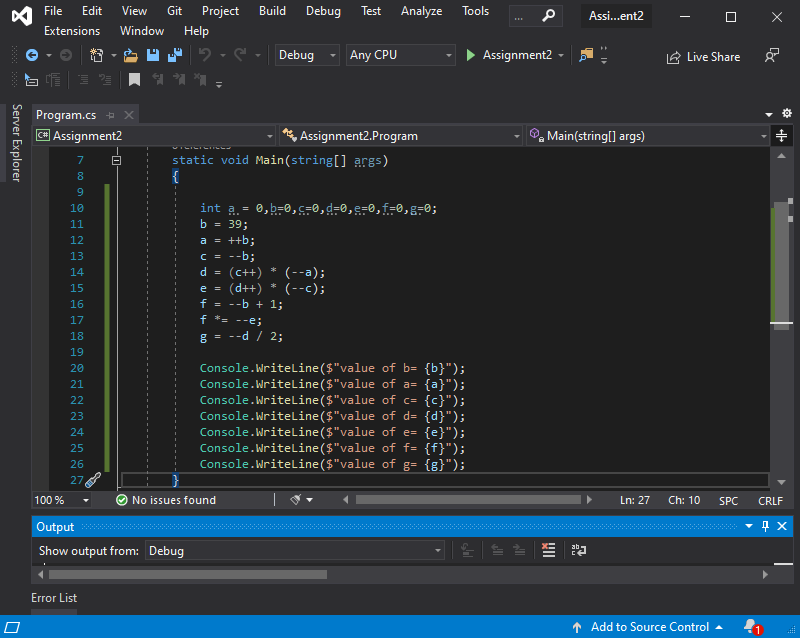
d = (c++)\*(--a);

e = (d++)\*(--c);

f = --b + 1;

f \*= --e;

g = --d / 2;

**Ans:** 

To understand the changing values of variables see table. I have shown each variable’s value after execution of line.Executing line numbers are shown with **respect to lines of screenshot attached**. (pre) and (post) shows that the variable’s value is either changed **before the execution of that particular line i.e (pre)** or **after the execution of particular line i.e (post).** Final values are highlighted with yellow color.

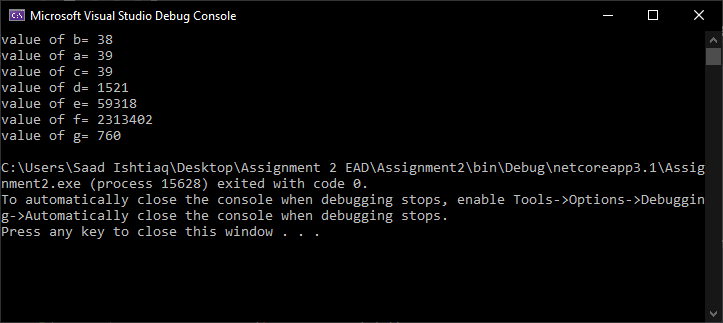
------------------------------------Executing Lines ----------------------🡪

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Variables** | **Line**  **10** | **Line**  **11** | **Line**  **12** | **Line**  **13** | **Line**  **14** | **Line**  **15** | **Line**  **16** | **Line**  **17** | **Line**  **18** |
| a | 0 |  | 40 |  | (pre)  39 |  |  |  |  |
| b | 0 | 39 | (pre)  40 | (pre)  39 |  |  | (pre)  38 |  |  |
| c | 0 |  |  | 39 | (post)  40 | (pre)  39 |  |  |  |
| d | 0 |  |  |  | 1521 | (post)  1522 |  |  | (pre)  1521 |
| e | 0 |  |  |  |  | 59319 |  | (pre)  59318 |  |
| f | 0 |  |  |  |  |  | 39 | 2313402 |  |
| G | 0 |  |  |  |  |  |  |  | 760 |

---------------------------------Values Assigned-----------------------------🡪

**Console Output:**

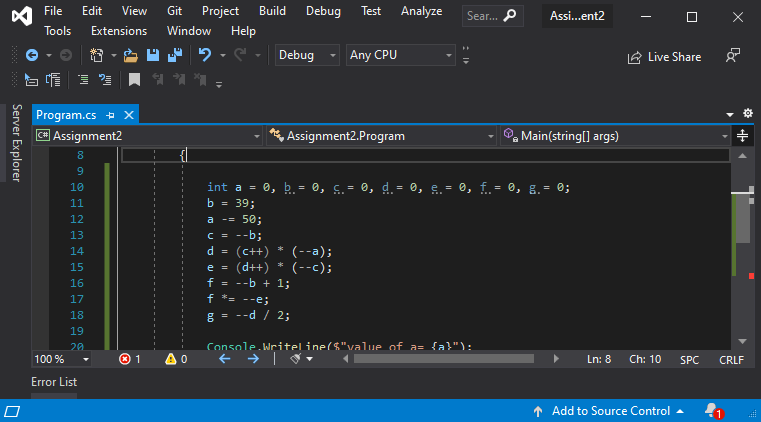
It shows final values matches.

****

**(B)**

If a becomes a- =50 which variables values will be affected? Please write down their changed values.

**Ans:**



Value of “a” is changed here

Same explanantion of solution as wriiten above.

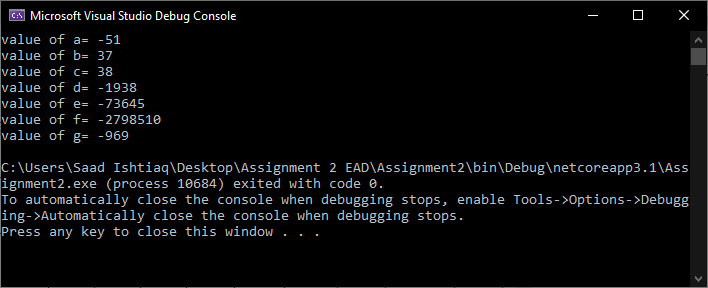
------------------------------------Executing Lines ----------------------🡪

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Variables** | **Line**  **10** | **Line**  **11** | **Line**  **12** | **Line**  **13** | **Line**  **14** | **Line**  **15** | **Line**  **16** | **Line**  **17** | **Line**  **18** |
| a | 0 |  | -50 |  | (pre)  -51 |  |  |  |  |
| b | 0 | 39 |  | (pre)  38 |  |  | (pre)  37 |  |  |
| c | 0 |  |  | 38 | (post)  39 | (pre)  38 |  |  |  |
| d | 0 |  |  |  | -1938 | (post)  -1937 |  |  | (pre)  -1938 |
| e | 0 |  |  |  |  | -73644 |  | (pre)  -73645 |  |
| f | 0 |  |  |  |  |  | 38 | -2798510 |  |
| G | 0 |  |  |  |  |  |  |  | -969 |

---------------------------------Values Assigned-----------------------------🡪

**Console Output:**

It shows final values matches.



**Q3:** Logical operators like &,|,-,^ etc. are also known as bitwise operators. If you want to do AND operation between two numbers like 8(1000) and 5(0101) then its output will be 0.Write a program that takes first argument as the size of array and remaining arguments as numbers from command line arguments. Select two numbers randomly from the arguments numbers and apply bitwise operations using switch statement. Finally display the result in proper output formatting. You can use loop as well to operate your function multiple times.

Note: Do Type checking and Exceptional handling as well

**Ans: Run :** dotnet run 5 1 2 3 4 5

using System;

namespace Assignment2

{

class Program

{

static void Main(string[] args)

{

try

{

int size = int.Parse(args[0]);

int[] array = new int[size];

int i = 0;

while (i < size)

{

array[i] = int.Parse(args[i + 1]);

i++;

}

//Generating Random Number

int randomNumber1 = (new Random()).Next(0, array.Length);

int num1=array[randomNumber1];

int randomNumber2 = (new Random()).Next(0, array.Length);

int num2=array[randomNumber2];

string[] operators = { "^", "|", "&", "~"};

foreach(string operations in operators)

{

string message = operations switch

{

"^"=> $"{num1,-3} ^ {num2,3} = {num1^num2,5}",

"|"=> $"{num1,-3} | {num2,3} = {num1|num2,5}",

"&"=> $"{num1,-3} & {num2,3} = {num1&num2,5}",

"~"=> $"~{num1,-8} = {~num1,5} \n~{num2,-8} = {~num2,5}",

\_=>"default"

};

Console.WriteLine(message);

}

}

catch(System.FormatException ex)

{

Console.WriteLine($"Invalid Input. {ex.Message}");

}

catch(Exception e)

{

Console.WriteLine($"{e.GetType()}. -> {e.Message}");

}

}

}

}

**Q4:** Write a program that takes size of student subjects and their numbers in command line arguments. Apply Switch expression to display given statements with proper output formatting:

**Ans: Run :** dotnet run 11 -9 40 55 60 65 69 70 76 83 95 102

using System;

namespace Assignment2

{

class Program

{

static void Main(string[] args)

{

try

{

int size = int.Parse(args[0]);

int[] array = new int[size];

int i = 0;

while (i < size)

{

array[i] = int.Parse(args[i + 1]);

i++;

}

foreach (int number in array)

{

string message = number switch

{

\_ when ((number < 50)) => "GPA: 0.0 Fail",

\_ when ((number > 50 && number <57 )) => "GPA: 1.0",

\_ when ((number >= 57 && number <64 )) => "GPA: 1.7",

\_ when ((number >= 64 && number <= 67)) => "GPA: 2.0",

\_ when ((number >= 68 && number < 70)) => "GPA: 2.7",

\_ when ((number >= 70 && number < 75)) => "GPA: 3.0",

\_ when ((number >= 75 && number < 80)) => "GPA: 3.4",

\_ when ((number >= 80 && number < 85)) => "GPA: 3.7",

\_ when ((number >= 85 && number <= 100)) => "GPA: 4.0",

\_ =>"Not Valid Marks"

};

Console.WriteLine($"Your Marks:{number,3} --------- You Got {message,3}");

};

}

catch (System.FormatException ex)

{

Console.WriteLine($"Invalid Input. -> {ex.Message}");

}

catch(Exception e)

{

Console.WriteLine($"{e.GetType()}. -> {e.Message}");

}

}

}

}

**Q5:** Write a program that takes a floating-point number from user as input and then rounds off the number. Rounding the number should also be based on choice from user. Provide two methods of rounding, One with the banker’s algorithm and the other one with the old method of rounding.

**Ans:**

using System;

namespace Assignment2

{

class Program

{

static void Main(string[] args)

{

float number = default;

int choice = default;

try

{

Console.WriteLine("Enter Number: ");

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

Console.WriteLine("Which method of round off do you want? Select 1 / 2");

Console.WriteLine("1- Banker's Algorithm");

Console.WriteLine("2- Traditional Method");

Console.WriteLine("Your Choice: ");

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

if (choice==1)

{

number = System.Convert.ToInt32(number);

}

else if(choice==2)

{

number= (float)(Math.Round(value: number, digits: 0, mode: MidpointRounding.AwayFromZero));

}

else

{

Console.WriteLine("Your selected incorrect option!");

Console.WriteLine("Number is not rounded off");

}

Console.WriteLine($"Your Answer: {number}");

}

catch(Exception e)

{

Console.WriteLine($"{e.GetType()}. -> {e.Message}");

}

}

}

}

**Q6:** Differentiate between implicit and explicit type casting with examples. Why explicit type casting is not good?

**Ans:**

|  |  |
| --- | --- |
| **Implicit** | **Explicit** |
| Implicit type casting is performed by the compiler on its own when it encounters a mixed data type expression in the program. | Explicit type casting is performed by the programmer. In this type casting programmer tells compiler to type cast one data type to another data type using type casting operator. |
| It is also known as automatic conversion as it is done by compiler without programmer’s assistance. implicit casting doesn’t require a casting operator. | There is some risk of information loss is there, so one needs to be careful while doing it. |
| Safe to use. | Isn’t safe, as data is being lost. |
| **Examples:**  //implicit  int a = 10;  double b = a; // an int can be safely cast into a double    Console.WriteLine(b); | **Examples:**  //explicit  double c = 9.8;  //int d = c; // compiler gives an error for this line    int d = (int)c; // d is 9 losing the .8 part  Console.WriteLine(d); |

Explicit type casting is not good because it is not convenient and more over there is threat of **data lost** as shown in above example.

**Q7:** Please find out the errors and explain the valid reasons in the snippet if it has.

namespace ConsoleApp2

{

public class Model1

{

private protected int firstVariable;

private int secondVariable;

}

public class Model2 : Model1

{

internal protected int thirdVariable;

public int fourthVariable;

}

public class Model3 : Model2

{

protected int fifthVariable;

}

public class Model4 : Model3

{

public void Driver()

{

Model3 modelObj = new Model3();

modelObj.fifthVariable = 50; // error: You cannot access private data member of a class by its own object!!!

fifthVariable = 23;

}

int sixthVariable;

}

class Program

{

static void Main(string[] args)

{

Model2 medel2\_Obj = new Model2();

Model4 model4\_Obj = new Model4();

Model4 model4\_Obj2 = new Model3(); // error: derived class cannot make object of base class.

model4\_Obj.secondVariable; // error : secondVariable is private data member, it cannot be accessed here.

model4\_Obj.fourthVariable; // error: uninitiallized variable

medel2\_Obj.firstVariable; //error: private protected can only be accessed within its own class or derived class object outside of class.

}

}

}

namespace ConsoleApp1

{

class Program : Model2

{

static void Main(string[] args)

{

Model1 obj = new Model1();

Program obj\_2 = new Program();

obj\_2.thirdVariable=1;

obj.firstVariable=4; //error: you cannot access private protected of one application in other.But If it is in same application then it must be accessed by object of derived class.

//(ambigous piece of code, I don’t know either ConsoleApp1 is in same project or different. Yet wrote both the cases.)

}

}

}