A logo for a company

Description automatically generated

NAME: **TADIWANASHE KAPFIDZE**

REG NUMBER: **H230762X**

NAME: **NIGEL MUPIRA**

REG NUMBER: **H230828V**

PROGRAM: **INFORMATION SECURITY AND ASSURANCE**

**SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY**

MODULE: **VISUAL PROGRAMMING CONCEPTS**

LECTURER: **MR S. MAVUCHI**

**GROUP ASSIGNMENT 1**

1. State and explain the .Net framework and give the benefits of each component.

The .Net framework is a software platform developed by Microsoft. It provides tools, libraries and runtime environments required to develop web applications, mobile applications and gaming applications. Its components include:

1. **Common Language Runtime - i**t is a runtime environment that provides services such as memory management, exception handling and security management. Benefits include:

* allocation of memory is made easier because of the memory management through garbage collection
* exception handling techniques help to write authentic and robust code

1. **Class Library - i**t is a set of pre-built classes and types that can be reused. It helps in increasing the development process and providing access to the functionality of a system.Benefits include:

* reduced development time due to the pre-defined classes
* promotion of code reusability

1. **ASP.NET - i**t is a framework that is used to build web applications and services. Benefits include:

* easy to customization because it is integrated into the environment of a windows server
* it makes use of reusable code making it very easy to maintain

1. Differentiate a rectangular array from a ragged array with the aid of examples.

|  |  |  |
| --- | --- | --- |
| Characteristic | Rectangular array | Ragged array |
| Structure | all rows contain the same number of elements, and all columns also contain the same number of elements | rows have different lengths forming an irregular structure |
| Storage | the elements are stored in a single memory block | each row is stored in its own memory block and the reference to each row is stored in the main array |
| Array accessing | indices are used to access the elements, it is efficient | two levels of indexing are required to access elements |

Example of rectangular array:

int[ , ] squareMatrix = new int[2, 2] {{0, 1}, {3, 7}};

Example of ragged array:

int[ ][ ] jaggedArray = new int[3][ ];

jaggedArray[0] = new int[ ] {11, 22, 33, 44};

jaggedArray[1] = new int[ ] {1, 2, 3};

jaggedArray[2] = new int[ ] {5, 10, 15, 20, 24};

1. Discuss the notion that C# is a type-safe, modern and interoperable language.

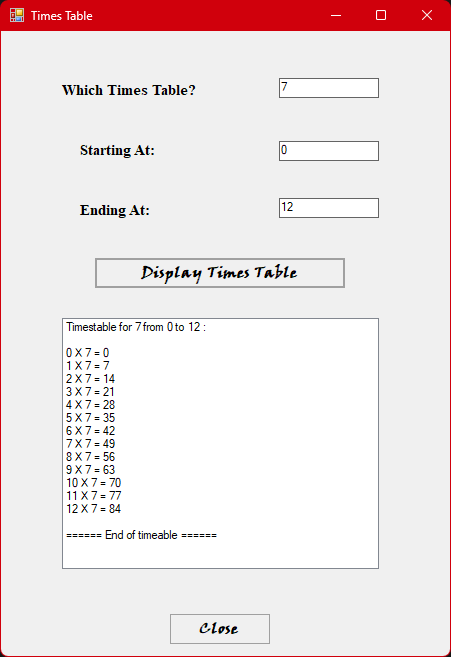
**Type-safety** - all variables and expressions in C# must have a well-defined data type at compile-time. This ensures that type errors are identified early in the development process, reducing the possibility of runtime errors. Type safety promotes code reliability and helps in the detection of errors during the development stage.

**Modernity** - C# is an object-oriented programming language, it allows the code to be organized into classes, encapsulate data, and define relationships between objects.

Because C# allows generics, building algorithms and data structures that can handle a variety of data types without compromising type safety makes it possible to write reusable code. C# is also constantly evolving with new features added in every major release keeping the language up-to-date with modern programming trends and paradigms.

**Interoperability** – C# “plays well with others”. It is highly interoperable as it uses the common language runtime that allows the code to run on multiple platforms. C# also seamlessly integrates with other languages on the .NET platform, this allows developers to leverage existing codebases and libraries written in different languages.

1. Interface:



Code:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace timesTable

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

// setting form initial position to the very center of the screen

StartPosition = FormStartPosition.CenterScreen;

}

private void button1\_Click(object sender, EventArgs e)

{

// accepting user input for timestable number

int num=Convert.ToInt32(textBox1.Text);

// error handling for negative timestable number

if (num <= 0)

{

MessageBox.Show("The times table number must be positive.");

return;

}

// determining starting and end point of timestable from input

int start\_num = Convert.ToInt32(textBox2.Text);

int end\_num = Convert.ToInt32(textBox3.Text);

// ensuring that starting point is lower than the end point

if (start\_num >= end\_num)

{

MessageBox.Show("Please enter a valid range for the starting and ending number.");

return;

}

// clearing listbox before displaying timestable

listBox1.Items.Clear();

// opening statement

listBox1.Items.Add("Timestable for " + num + " from " + start\_num + " to " + end\_num + " :");

// empty line between opening statement and timestable

listBox1.Items.Add(" ");

// the loop that calculates and displays our timestable

for (int i = start\_num; i <= end\_num; i++)

{

listBox1.Items.Add(i + " X " + num + " = " + i \* num);

}

// empty line between timestable and closing statement

listBox1.Items.Add(" ");

// closing statement

listBox1.Items.Add("====== End of timeable ======");

}

private void button2\_Click(object sender, EventArgs e)

{

//adding a button to close the form window

this.Close();

}

}

}