Before reading past below instructions:

1. Create an account in Github using your name in this format: lastname\_firstname\_section
2. Request access to [Lycevm<3Alabang · GitHub](https://github.com/Lycevm-3Alabang)
3. Upload this file ON YOUR GITHUB ACCOUNT with answer under the title / file name : E3\_Assessment\_\_[Section]\_[LastnameFirstName]  
   example: E3\_Assessment\_\_BSCS32E1\_AlamoNinoFrancisco

Help: [Get started with GitHub documentation - GitHub Docs](https://docs.github.com/en/get-started)

**Sample Assessment for Introduction to Programming**

This assessment is designed to evaluate your understanding of basic programming concepts in C#, HTML, CSS, and JavaScript.

Instructions: Read each question carefully and provide complete and clear answers. Avoid multiple-choice format responses. Focus on demonstrating your understanding through code, explanations, and discussions.

**Part 1: C# (30 points)**

(10 points) Write a C# program that calculates the area of a triangle given its base and height. Include user input for both values and display the calculated area.

using System;

class triangle

{

static void Main()

    {

        Console.WriteLine("Enter the base of the triangle: ");

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

        Console.WriteLine("Enter the height of the triangle: ");

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

        double area = 0.5 \* Length \* base;

        Console.WriteLine($"The area of the triangle is: {area}");

    }

}

**(10 points) Declare an array of 5 integers and fill it with values based on a user-defined formula (e.g., n^2). Then, print the largest element in the array.**

using System;

namespace Array

{

    class Compare

    {

        static void Main(string[] args)

        {

            int[] Arr1 = new int[5];

            Console.WriteLine("Enter the elements of the array:");

            for (int i = 0; i < Arr1.Length; i++)

            {

                Console.Write($"Element {i + 1}: ");

                Arr1[i] = int.Parse(Console.ReadLine());

            }

            Console.WriteLine("Numbers within the Array:");

            foreach (int num in Arr1)

            {

                Console.WriteLine(num);

            }

            int bigger = Arr1[0];

            for (int i = 1; i < Arr1.Length; i++)

            {

                if (Arr1[i] > bigger)

                {

                    bigger = Arr1[i];

                }

            }

            Console.WriteLine("The largest element in the array is: " + bigger);

            Console.ReadKey();

        }

    }

}

**(10 points) Implement a simple for loop that iterates from 1 to 10 and prints each number along with its square root.**

using System;

class loop

{

    static void Main()

    {

        for (int i = 1; i <= 10; i++)

        {

            double squareRoot = Math.Sqrt(i);

            Console.WriteLine($"Number: {i}, Square Root: {squareRoot}");

        }

    }

}

**Part 2: HTML, CSS, and JavaScript (30 points)**

**HTML (10 points):** You are provided with the following incomplete HTML code snippet:

**HTML**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>My Website</title>**

**</head>**

**<body>**

**<h1>Welcome to...</h1>**

**<p>This is a paragraph...</p>**

**<ul>**

**<li>Item 1</li>**

**<li>Item 2</li>**

**</ul>**

**</body>**

**</html>**

Complete the code snippet by adding the following elements:

An image within the <body> tag with a relevant src attribute.

An ordered list (<ol>) with three items.

A hyperlink within a <p> tag that points to an external website.

A CSS styling rule using an inline style attribute to change the font color of the <h3> heading.

<!DOCTYPE html>

<html>

<head>

  <title>My Website</title>

  <link rel="stylesheet" type="text/css" href="style.css">

  <script src="script.js"></script>

</head>

<body>

  <h1>Welcome to...</h1>

  <p>This is a paragraph...</p>

  <ul>

    <li>Item 1</li>

    <li>Item 2</li>

    <li>Item 3</li>

  </ul>

  <img src="youtube.jpg" alt="YouTube logo">

  <p><a href="https://www.youtube.com">YouTube</a></p>

  <h1>Header 1</h1>

  <h2>Header 2</h2>

  <h3 style="color: red;">Header 3</h3>

  <label for="numberInput">Enter a number:</label>

  <input type="number" id="numberInput">

  <button onclick="EvenOrOdd()">Check Number</button>

  <p id="resultParagraph"></p>

</body>

</html>

CSS (10 points): Create a CSS stylesheet that defines the following styles:

Change the background color of the body element to light blue.

Apply a padding of 20px to all headings (h1, h2, h3).

Set the font size of the <p> tag to 14px.

Make the list items (li) have a bullet point style instead of the default numbers.

body {

    font-family: Arial, sans-serif;

    background-color: lightblue;

  }

  h1, h2, h3 {

    padding: 20px;

  }

  p {

    font-size: 14px;

  }

  li {

    list-style-type: disc;

  }

  img {

    max-width: 100%;

    height: auto;

  }

**JavaScript (10 points):** Write a JavaScript function that takes a number as input and returns a string indicating whether the number is even or odd. Then, add a button to your HTML page that, when clicked, calls this function and displays the result (even or odd) in a paragraph element below the button.

function EvenOrOdd() {

    var inputNumber = parseInt(document.getElementById("numberInput").value);

    var result = checkEvenOrOdd(inputNumber);

    document.getElementById("resultParagraph").innerText = `The number is ${result}.`;

}

function checkEvenOrOdd(number) {

    if (number % 2 === 0) {

        return "Even";

    } else {

        return "Odd";

    }

}

**Part 3: Essay Question (40 points)**

Discuss the importance of object-oriented programming (OOP) concepts in software development. Explain the key principles of OOP (encapsulation, inheritance, polymorphism, abstraction) and provide examples of how they can be used to create more efficient, maintainable, and reusable code. Include real-world scenarios or cases where OOP is particularly valuable.

Object-Oriented Programming is known for its simplicity and efficiency. It breaks down complex systems into manageable parts to objects, making code easier to understand and maintain. OOP's flexibility and extensibility allow developers to extend existing functionality without altering the existing codebase, promoting adaptability.

One of the key principles of OOP is encapsulation, where data and methods are bundled into a single unit, known as a class. This enhances data security and organization, as internal details are hidden from external access. Abstraction is another vital principle, simplifying complex systems by exposing only relevant details, making the code more user-friendly and accessible.

Another is Inheritance, it enables the creation of common properties and methods in a base class, which are then inherited by specific types or subclasses. This promotes a hierarchical structure, aiding in code organization and reuse. Polymorphism, a versatile feature, allows for the flexible handling of different types of objects, supporting seamless interactions within the code.

In everyday situations, Object-Oriented Programming (OOP) proves useful. Think about a bank's computer system. OOP helps organize things like bank accounts and customer details. It makes sure that data is kept safe, and transactions can happen smoothly and quickly.

Points Distribution:

Each part carries equal weight (30 points).

Code clarity, functionality, and explanations will be considered in grading.

The essay question focuses on understanding and application of OOP concepts.