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;

namespace TriangleAreaCalculator

{

class Program

{

static void Main(string[] args)

{

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

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

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

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

double area = 0.5 \* b \* h;

Console.WriteLine($"The area of the triangle with base {b} and height {h} 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.**

**Sorry po, hindi ko po masyadong maintindihan yung tanong.**

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

using System;

namespace LoopwithSqrt

{

class Program

{

static void Main(string[] args)

{

Console.Write("Loop with sqrt\n");

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

{

double sqrt = Math.Sqrt(i);

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

}

}

}

}

**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>

  <link rel="stylesheet" href="style.css">

  <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>

  <img src="img/woah.jpg">

  <ol>

    <li>me</li>

    <li>myself</li>

    <li>and i</li>

  </ol>

  <p>This is a <a href="https://github.com/SalazarGabrielAngelo-BSIT32E1">hyperlink</a> my github profile.</p>

  <h3 style="color: blue;">This is a styled heading</h3>

</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 {

    background-color: lightblue;

  }

  h1, h2, h3 {

    padding: 20px;

  }

  p {

    font-size: 14px;

  }

  li {

    list-style-type: disc;

  }

**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.

<!DOCTYPE html>

<html>

<head>

  <title>My Website</title>

</head>

<body>

  <script src = "odd&even.js"></script>

  <h1>Even or Odd Checker</h1>

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

  <input type="number" id="numInput">

  <button onclick="displayResult()">Check</button>

  <p id="result"></p>

</body>

</html>

function checkEvenOrOdd(number) {

    if (number % 2 === 0) {

      return "Even";

    } else {

      return "Odd";

    }

  }

  function displayResult() {

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

    var result = checkEvenOrOdd(inputNumber);

    document.getElementById("result").innerText = "The number is " + result + ".";

  }

**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 (OOP) is a programming paradigm that revolves around the concept of "objects," which represent real-world objects to represent and manipulate data, including the actions that can be performed. OOP brings several key principles to software development, these principles contribute to developing an efficient code and maintainability of the software.

* **Encapsulation:** Think of encapsulation like a treasure chest. The chest holds valuable items (data) and is locked with a key (methods). Only certain methods (key holders) can unlock and access the treasure inside.
* **Inheritance:** Inheritance is like passing down traits from parents to children. A child inherits characteristics like eye color or height from their parents. Just like in programming, a subclass(child) can inherit properties and behavior from a superclass(parent).
* **Polymorphism:** Polymorphism is like a shapeshifter. It allows objects to take on different forms or behaviors depending on the context. For example, a shapeshifter might transform into a bird to fly or a fish to swim. In programming, polymorphism enables objects of different types to be treated as if they are the same type.
* **Abstraction:** Abstraction is like using a TV remote without knowing how it works internally. You press a button, and the TV responds accordingly, without needing to understand the complex electronics inside. In programming, abstraction hides complex implementation details and focuses on what an object does rather than how it does it.

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.