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 Calculator

{

static void Main(string[] args)

{

double baseLength, height, area;

//get baseLength

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

if (!double.TryParse(Console.ReadLine(), out baseLength) || baseLength <= 0)

{

Console.WriteLine("Please enter a valid POSITIVE number for the base length.");

return;

}

// get height

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

if (!double.TryParse(Console.ReadLine(), out height) || height <= 0)

{

Console.WriteLine("Please enter a valid POSITIVE number for the height.");

return;

}

// Calculate area

area = 0.5 \* baseLength \* height;

Console.WriteLine($"Area of the triangle w/ base {baseLength} and height {height} is: {area}");

Console.ReadKey();

}

}

}

**(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 LargestElementInArray

{

class Compare

{

static void Main(string[] args)

{

int[] Arr1 = { 10, 5, 13, 7, 36 };

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;

namespace IterationLoop

{

class Loop

{

static void Main(string[] args)

{

int num = 1;

double sqr;

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

{

sqr = Math.Sqrt(num);

Console.WriteLine(num + " " + sqr);

num = num + 1;

}

}

}

}

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

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

**<!DOCTYPE html>**

**<html>**

**<head>**

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

**<style>**

**body {**

**background-color: lightblue;**

**}**

**h1, h2, h3 {**

**padding: 20px;**

**}**

**p {**

**font-size: 14px;**

**}**

**li {**

**list-style-type: disc;**

**}**

**</style>**

**</head>**

**<body>**

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

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

**<img src="sharkzilla.png" alt="Image drawn by me">**

**<ol>**

**<li>Duskblade of Markus</li>**

**<li>Essence Reaver</li>**

**<li>Markus' Ghostblade</li>**

**</ol>**

**<p>Visit for a<a href="https://www.youtube.com/watch?v=dQw4w9WgXcQ&pp=ygUTbmV2ZXJnb25uYWdpdmV5b3V1cA%3D%3D"> good time :)</a></p>**

**<h3 style="color: red;">Heading 3 is Red </h3>**

**<button onclick="checkNumber()">Check Number</button>**

**<p id="result"></p>**

**<script>**

**function checkNumber() {**

**var number = prompt("Enter number:");**

**if (isNaN(number)) {**

**document.getElementById("result").textContent = "Invalid input. Please enter a valid number.";**

**return;**

**}**

**number = Number(number);**

**if (number % 2 === 0) {**

**document.getElementById("result").textContent = number + " is even.";**

**} else {**

**document.getElementById("result").textContent = number + " is odd.";**

**}**

**}**

**</script>**

**</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.

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.

**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.

**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.

* OOP is integral in Software development due to how the principles of OOP is vast and can be applied in creating any application that needs repetitive functions and structuring. It enables different techniques to be used in ways that can progressively make the development process easier. Starting with Encapsulation, this allows you to accomplish things like controlling accessibility, an example for this is when you are able to hide a variable like ***balance*** in a banking system which can only be accessed through transaction without being able to directly change it.

Next is inheritance which allows data to literally be inherited by other variables, this makes data transfer easier since you will be creating a tree in which you can use to monitor the data transfer. Next is Polymorphism which refers to the ability of objects to take on multiple forms. It allows objects of different classes to be treated as objects of a common parent class, enabling a single interface to represent a general class of actions, for example is creating a program that defines shapes, you can use a single function that can do different things depending on the criteria that the user wants. Lastly is abstraction which means to make something vague in a program that can allow programmers to focus on what the object can do a function instead of thinking about how it does it, a great example of this is drawing shapes. An object can draw many shapes but they can be identified by what shape they are drawing.

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.