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 Program

{

static void Main(string[] args)

{

// Prompt the user to enter the base of the triangle

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

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

// Prompt the user to enter the height of the triangle

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

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

// Calculate the area of the triangle

double area = (baseLength \* height) / 2;

// Display the calculated area

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 ArrayFormulaExample

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter the value of n: ");

int n = Convert.ToInt32(Console.ReadLine());

int[] arr = new int[5];

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

{

arr[i] = n \* n;

n++;

}

int max = arr[0];

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

{

if (arr[i] > max)

{

max = arr[i];

}

}

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

}

}

}

**(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 SquareRootLoop

{

class Program

{

static void Main(string[] args)

{

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

{

double sqrt = Math.Sqrt(i);

Console.WriteLine("The square root of {0} is {1}", i, 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.

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.

**HTML:**

<!DOCTYPE html>

<html>

<head>

<title>My Website</title>

<link rel="stylesheet" type="text/css" href="1.css">

</head>

<body>

<h1>Welcome to My Website</h1>

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

<ol>

<li>Ordered Item 1</li>

<li>Ordered Item 2</li>

<li>Ordered Item 3</li>

</ol>

<ul>

<li>Item 1</li>

<li>Item 2</li>

</ul>\

<p>Visit <a href="https://www.example.com">Example Website</a></p>

<img src="image.jpg" alt="An image">

<h3>This is a heading</h3>

</body>

</html>

**CSS:**

body {

background-color: lightblue;

}

h1, h2, h3 {

padding: 20px;

}

p {

font-size: 14px;

}

li {

list-style-type: disc;

}

h3 {

color: blue;

}

**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.

**HTML:**

<!DOCTYPE html>

<html>

<head>

<title>Even or Odd</title>

</head>

<body>

<button id="checkButton">Check if number is even or odd</button>

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

<script src="2.js"></script>

</body>

</html>

**JS:**

function isEvenOrOdd(num) {

if (num % 2 === 0) {

return "even";

} else {

return "odd";

}

}

document.getElementById("checkButton").addEventListener("click", function() {

var num = prompt("Enter a number:");

var result = isEvenOrOdd(num);

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.

In software development, object-oriented programming, or OOP, is essential because it makes code more reusable, maintainable, and collaborative among developers. To achieve these advantages, the fundamental OOP concepts of encapsulation, inheritance, polymorphism, and abstraction are essential.

**Encapsulation:** By combining data and methods into a class, encapsulation provides abstraction and data security. This principle encourages information hiding and data integrity while shielding data from outside interference. For instance, a smartphone's internal parts are housed inside its body, making it possible for users to interact with the device through the user interface without being aware of its specific hardware.

**Inheritance:** Allows classes to inherit traits and behaviors from parent classes. With its hierarchical structure, code is well-organized and modifications to one section can be made without impacting the system as a whole. A "Child" class in a family tree representation, for example, may inherit traits from a "Parent" class.

**Polymorphism:** A common superclass or interface can be regarded as an object that consists of multiple classes thanks to polymorphism. With interchangeable implementations made possible by this flexibility, code maintenance is made easier and extensibility is encouraged. Based on the actual object type, Java's polymorphism enables dynamic method dispatch.

**Abstraction:** By focusing on what an object does rather than how it does it, abstraction models complex processes at a high level and makes them simpler. As a result, the code is more understandable and has less complexity. For instance, developers don't have to comprehend the intricate workings of the engine when working with an automobile object; instead, they can concentrate on critical features like braking and acceleration.

**Real-world scenarios where OOP is valuable include:**

* Mobile App Development: Encapsulation, inheritance, and polymorphism are key components of object-oriented programming (OOP), which enables the creation of modular, scalable, and maintainable applications.
* Game Development: OOP is used by game engines such as Unity to improve realism and effectively handle intricate game structures.
* Web development: Reusable code components are built into robust web applications using frameworks like Django and Ruby on Rails, which make use of OOP.

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.