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()

{

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

double baseLength = double.Parse(Console.ReadLine());

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

double height = double.Parse(Console.ReadLine());

double area = 0.5 \* baseLength \* height;

Console.WriteLine($"The area of the triangle with base {baseLength} and height{height} 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;

class Program

{

static void Main()

{

int[] array = new int[5];

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

{

array[i] = (i + 1) \* (i + 1); // n^2 formula

}

Console.WriteLine("Array elements:");

foreach (int num in array)

{

Console.WriteLine(num);

}

int maxElement = array[0];

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

{

if (array[i] > maxElement)

{

maxElement = array[i];

}

}

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

}

}

**(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 Program

{

static void Main(string[] args)

{

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.

**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.

By organizing actions and data into objects, object-oriented programming (OOP) facilitates the creation of software and makes code management and updates simpler. OOP promotes code reuse by utilizing inheritance and composition, enabling the creation of classes that extend or specialize existing ones. It facilitates the development of expandable systems by making things easily scalable to accommodate changing requirements. In OOP, abstraction hides specifics to highlight benefits and facilitate simpler comprehension and upkeep of the code. By limiting access to information, encapsulation protects data accuracy while enhancing code reliability. The interchangeability of objects is made possible via polymorphism, which also makes algorithm implementations simpler and increases code flexibility and reuse between classes.

• Encapsulation: combines methods and data into objects, concealing underlying workings and only exposing interfaces that are necessary.   
An example would be a Bank Account class that has get Balance, deposit, and withdraw methods.

• Inheritance: This promotes code reuse by allowing new classes subclasses to inherit traits and behaviors from more established classes super classes. Triangle, Rectangle, and Circle subclasses under the Shape superclass, for instance.

• Polymorphism: Allows methods to behave differently depending on the type of object by treating objects of different classes as belonging to a shared superclass. Subclasses implement the calculate Area method differently from the Shape superclass.

• Abstraction: Shows only the most important characteristics while hiding intricate implementation details.

Real-world scenarios or cases where OOP is particularly valuable:

• Software Development: By dividing systems into smaller objects, OOP is used in web development frameworks such as Ruby on Rails, Django, and Laravel to enable scalable and maintainable codebases.

• Game creation: OOP is widely used to model game entities, behaviors, and interactions in game creation. This allows for easy updating of existing code without affecting other components and allows for code reuse.

• Simulation and Modeling: OOP is useful in scientific research for simulating intricate systems, such biological or physical systems, as it makes experimentation and analysis simple by expressing each component as an entity with a state and behavior of its own.

• Development of Graphical User Interfaces (GUIs): Object-oriented programming (OOP) is widely used in GUI development to create interactive programs that represent windows, buttons, and menus as objects, making management and handling of user interaction easier.

• Embedded Systems: Especially in the automotive and aerospace industries, where modular and maintainable code is necessary for interacting with hardware components, object-oriented programming (OOP) is a must.

• Data Analysis and Visualization: As demonstrated by libraries like matplotlib and ggplot2, object-oriented programming (OOP) facilitates data analysis and visualization chores by grouping code into reusable objects, allowing for the effective investigation and sharing of insights from enormous datasets.

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.