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.**

**(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.**

**(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 Class

{

//first part

static void Triangle\_Area()

{

Console.Clear();

Console.Write("Enter Base: ");

float b = float.Parse(Console.ReadLine());

Console.Write("Enter Height: ");

float h = float.Parse(Console.ReadLine());

Console.WriteLine("Triangle Area: " + b \* h / 2);

Console.ReadKey();

}

//second part

static void Formula\_Array()

{

//declare new array

float[] array = new float[5];

Console.Clear();

Console.Write("Enter Starting Number: ");

array[0] = float.Parse(Console.ReadLine());

float max = array[0];

Console.Write("Enter Operand: ");

float n = float.Parse(Console.ReadLine());

while (true)

{

//rewrite incase wrong operation input

Console.Clear();

Console.WriteLine("Enter Starting Number: " + array[0]);

Console.WriteLine("Enter Operand: " + n);

Console.Write("Enter Operation (+, -, \*, /, ^): ");

switch (Console.ReadLine())

{

case "+":

for (int i = 1; i < 5; i++)

{

array[i] = array[i - 1] + n;

if (array[i] > max)

max = array[i];

}

Console.WriteLine("Array: " + string.Join(", ", array));

Console.WriteLine("Largest Number: " + max);

Console.ReadKey();

return;

case "-":

for (int i = 1; i < 5; i++)

{

array[i] = array[i - 1] - n;

if (array[i] > max)

max = array[i];

}

Console.WriteLine("Array: " + string.Join(", ", array));

Console.WriteLine("Largest Number: " + max);

Console.ReadKey();

return;

case "\*":

for (int i = 1; i < 5; i++)

{

array[i] = array[i - 1] \* n;

if (array[i] > max)

max = array[i];

}

Console.WriteLine("Array: " + string.Join(", ", array));

Console.WriteLine("Largest Number: " + max);

Console.ReadKey();

return;

case "/":

for (int i = 1; i < 5; i++)

{

array[i] = array[i - 1] / n;

if (array[i] > max)

max = array[i];

}

Console.WriteLine("Array: " + string.Join(", ", array));

Console.WriteLine("Largest Number: " + max);

Console.ReadKey();

return;

case "^":

for (int i = 1; i < 5; i++)

{

array[i] = (float)Math.Pow(array[i - 1], n);

if (array[i] > max)

max = array[i];

}

Console.WriteLine("Array: " + string.Join(", ", array));

Console.WriteLine("Largest Number: " + max);

Console.ReadKey();

return;

default:

//loop back when operation is invalid

break;

}

}

}

//third part

static void Square\_Root()

{

Console.Clear();

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

{

Console.WriteLine("Number: " + i + ", Square Root: " + Math.Sqrt(i));

}

Console.ReadKey();

}

//main method

static void Main(string[] args)

{

//select program

while (true)

{

Console.Clear();

Console.WriteLine("Select Program");

Console.WriteLine("1 Triangle Area");

Console.WriteLine("2 Formula Array");

Console.WriteLine("3 Square Root");

Console.WriteLine("4 Exit");

switch (Convert.ToInt32(Console.ReadLine()))

{

case 1:

Triangle\_Area();

break;

case 2:

Formula\_Array();

break;

case 3:

Square\_Root();

break;

case 4:

//exit

return;

default:

break;

}

}

}

}

**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.

HTML

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="stylesheet.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="image.png"/>

<ol>

<li>Item 3</li>

<li>Item 4</li>

<li>Item 5</li>

</ol>

<p>I have learned that <a href="https://www.youtube.com/watch?v=GJDNkVDGM\_s">there is no need to be upset</a></p>

<h3 style="color: red">h3 heading</h3>

<input type="text" id="number" oninput="this.value=this.value.replace(/(?![0-9])./gmi,'')"/>

<button onclick="Function()">calculate</button>

<p id="output"></p>

<script>

function Function(){

document.getElementById("output").innerHTML = document.getElementById("number").value % 2 == 0 ? "even" : "odd";

}

</script>

</body>

</html>

CSS

body {

background-color: lightblue;

}

h1, h2, h3 {

padding: 20px

}

p {

font-size: 14px

}

li {

list-style-type: disc;

}

**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 revolves around the concept of "objects," which can contain data in the form of fields, and code, in the form of methods. OOP concepts like encapsulation, inheritance, polymorphism, and abstraction are fundamental principles that facilitate the design, implementation, and maintenance of software systems.

**Encapsulation:**

Encapsulation bundles data and methods within a single unit, such as a class, hiding internal details and exposing only necessary interfaces. A real-life example would be a car. The car object encapsulates various attributes like speed, fuel level, and methods like accelerate() and brake(). These details are hidden from the outside, and users interact with the car through a limited set of interfaces like pressing the accelerator or brake pedal.

**Inheritance:**

Inheritance allows a new class to inherit properties and behaviors from an existing class, promoting code reuse. A real-life example is the relationship between different types of vehicles. A base class "Vehicle" may have attributes like speed and methods like start() and stop(). Derived classes like "Car" and "Truck" inherit these attributes and methods but may also have additional features specific to them, like "Car" having a convertible roof or "Truck" having a loading bed.

**Polymorphism:**

Polymorphism allows objects of different classes to be treated as objects of a common superclass. An example is a shape hierarchy in a drawing application. Various shapes like circles, rectangles, and triangles can all be treated as instances of a common "Shape" superclass. Each shape may have a different implementation of a method like calculateArea(), allowing the same code to work with different shapes interchangeably.

**Abstraction:**

Abstraction involves representing essential features while hiding irrelevant details. A real-life example is a remote control. Users interact with the remote control using buttons without needing to know the internal workings of the devices it controls, like a TV or a DVD player. The remote control abstracts away the complexities of each device, providing a simplified interface for interaction.

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.