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)

{

double baseLength, height;

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

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

{

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

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

}

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

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

{

Console.WriteLine("Invalid input. Please enter a valid positive number for the height.");

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

}

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(string[] args)

{

int[] numbers = new int[5];

Console.WriteLine("Enter 5 integers to fill the array:");

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

{

Console.Write($"Enter value {i + 1}: ");

while (!int.TryParse(Console.ReadLine(), out numbers[i]))

{

Console.WriteLine("Invalid input. Please enter an integer.");

Console.Write($"Enter value {i + 1}: ");

}

}

int max = numbers[0];

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

{

if (numbers[i] > max)

{

max = numbers[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;

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Number\t\tSquare Root");

Console.WriteLine("============================");

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

{

double squareRoot = Math.Sqrt(i);

Console.WriteLine($"{i,-10}\t\t{squareRoot,10:F4}");

}

}

}

**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>

</head>

<body>

<h1>Welcome</h1>

<p>Unordered List</p>

<ul>

<li>Pogi</li>

<li>Ako</li>

</ul>

<img src="pogi.jpg" alt="sobrang\_pogi">

<ol>

<p>Ordered List</p>

<li>Ang</li>

<li>Pogi</li>

<li>Ko</li>

</ol>

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

<h3 style="color: blue;">Heading 3</h3>

</body>

</html>

**CSS:**

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>Even or Odd Checker</title>

<style>

#resultParagraph {

font-size: 18px;

color: #333;

font-weight: bold;

margin-top: 10px;

}

</style>

</head>

<body>

<h1>Even or Odd Checker</h1>

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

<input type="number" id="numberInput">

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

<p id="resultParagraph"></p>

<script>

function checkEvenOrOdd() {

var number = parseInt(document.getElementById("numberInput").value);

if (number % 2 === 0) {

document.getElementById("resultParagraph").innerText = "The number is even.";

} else {

document.getElementById("resultParagraph").innerText = "The number is odd.";

}

}

</script>

</body>

</html>

**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.

**Answer:**

**Encapsulation**: Encapsulation involves bundling data and methods that operate on the data into a single unit or class. This helps in hiding the internal implementation details of an object and allows access to data only through well-defined interfaces. For example, a Car class might encapsulate properties like speed and methods like accelerate() and brake(). Users interact with the car object through these methods, without needing to know how they are implemented.

**Inheritance:** Inheritance allows a class (subclass) to inherit properties and behaviors from another class (superclass). This promotes code reuse and helps in creating a hierarchy of classes with shared characteristics. For instance, a Vehicle class might have properties and methods common to all vehicles, and specific vehicle types like Car and Truck can inherit from it, gaining those common functionalities.

**Polymorphism**: Polymorphism allows objects of different classes to be treated as objects of a common superclass. This enables code to be written more generically, as it can operate on objects of various types without needing to know their specific classes. For example, a draw() method in a superclass Shape can be overridden in subclasses like Circle and Rectangle to provide different implementations, but they can all be treated uniformly when invoked through the superclass reference.

**Abstraction:** Abstraction involves hiding the complex implementation details and showing only the essential features of an object. This simplifies the programming model and reduces complexity, making code more understandable and maintainable. For instance, a BankAccount class might abstract away the intricacies of banking operations, exposing methods like deposit() and withdraw() to interact with the account.

OOP principles provide a structured approach to software development, leading to more efficient, maintainable, and reusable code.

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.