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.

**Answer:**

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 = CalculateTriangleArea(baseLength, height);

Console.WriteLine("The area of the triangle is: " + area);

}

static double CalculateTriangleArea(double baseLength, double height)

{

return 0.5 \* baseLength \* height;

}

}

**(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.**

**Answer:**

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 max = array[0];

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

{

if (array[i] > max)

{

max = array[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.**

**Answer:**

using System;

class Program

{

static void Main()

{

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

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

{

double squareRoot = Math.Sqrt(i);

Console.WriteLine($"{i}\t{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.

**ANSWER:**

<!DOCTYPE html>

<html>

<head>

<title>My Website</title>

</head>

<body>

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

<h3 style= ”color: blue;”> This is a blue heading</h3>

<p>This is a paragraph with a <a href=<https://example.com>>hyperlink to an external website</a></p>

<ul>

<li>Item 1</li>

<li>Item 2</li>

</ul>

<img src= “example.jpg” alt= “Image” />

<ol>

<li>Item 1</li>

<li>Item 2</li>

<li>Item 3</li>

</ol>

</body>

</html>

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.

**ANSWER:**

/\* Style.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.

**Answer:**

<!DOCTYPE html>

<html>

<head>

<title>Even/Odd Checker</title>

</head>

<body>

<button onclick="checkEvenOdd()">Check Even/Odd</button>

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

<script>

function checkEvenOdd() {

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

var resultParagraph = document.getElementById("resultParagraph");

if (isNaN(number)) {

resultParagraph.textContent = "Invalid input. Please enter a valid number.";

} else {

if (number % 2 === 0) {

resultParagraph.textContent = number + " is even.";

} else {

resultParagraph.textContent = 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:**

Object oriented programming (OOP) helps in creating software by grouping data and actions into objects making it easier to manage and update code. By using inheritance and composition OOP encourages the recycling of code allowing for the development of classes that build upon or specialize ones. It supports the creation of expandable systems by making it simple to adjust and expand objects to meet changing needs. Abstraction, in OOP conceals details focusing on features and making code understanding and maintenance easier. Encapsulation safeguards data accuracy by restricting access, to information improving code dependability. Polymorphism allows for object interchangeability enhancing code flexibility and reuse across classes while simplifying algorithm implementations.

**The Key Principles Of OOP:**

* **Encapsulation:** Bundles data and methods into objects, hiding internal details and exposing only necessary interfaces.

Example: BankAccount class with methods like deposit(), withdraw(), and getBalance().

* **Inheritance:** Allows new classes (subclasses) to inherit properties and behaviors from existing classes (superclasses), promoting code reuse.

Example: Shape superclass with Circle, Rectangle, and Triangle subclasses.

* **Polymorphism:** Enables objects of different classes to be treated as objects of a common superclass, allowing methods to behave differently based on the object type.

Example: Shape superclass with calculateArea() method implemented differently in subclasses.

* **Abstraction:** Hides complex implementation details, exposing only essential features.

Example: Car class with simple methods like start(), stop(), etc., hiding internal engine details.

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

* **Software Development:** OOP is employed in web development frameworks like Ruby on Rails, Django, and Laravel, enabling scalable and maintainable codebases by breaking down systems into smaller objects.
* **Game Development:** In game development, OOP is extensively used to model game entities, behaviors, and interactions, facilitating code reuse and easy modification without disrupting other components.
* **Simulation and Modeling:** OOP is valuable in scientific research for modeling complex systems such as physical systems or biological processes, allowing easy experimentation and analysis by representing each component as an object with its own state and behavior.
* **Graphical User Interface (GUI) Development:** OOP is common in GUI development for creating interactive applications where elements like windows, buttons, and menus are represented as objects, simplifying management and user interaction handling.
* **Embedded Systems:** OOP is essential in embedded systems programming, particularly in industries like automotive and aerospace, where modular and maintainable code is crucial for interacting with hardware components.
* **Data Analysis and Visualization:** OOP aids in data analysis and visualization tasks by organizing code into reusable objects, as seen in libraries like matplotlib and ggplot2, enabling efficient exploration and communication of insights from large 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.