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;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Assessment\_E3

{

class Program

{

static void Main(string[] args)

{

//AREA OF A TRIANGLE

Console.WriteLine("AREA OF TRIANGLE");

double num = 0.5; // i converted the 1/2 into decimal

Console.Write("Please Enter Height: "); // user input

int height = Convert.ToInt32(Console.ReadLine()); // convert string input into an integer

Console.Write("Please Enter base: "); // user input

int Base = Convert.ToInt32(Console.ReadLine()); // convert string input into an integer

double area = num \* height \* Base; // formula to get area of triangle

Console.WriteLine("The area of a triangle is: " + area); // result

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Console.WriteLine("ARRAY OF 5 INTEGERS");

// Declare an array of 5 integers

int[] myArray = new int[5];

// Fill the array with values based on n^2

for (int i = 0; i < 5; i++)

{

Console.Write($" Please enter a number {i + 1}: ");

int userInput;

if (int.TryParse(Console.ReadLine(), out userInput))

{

myArray[i] = userInput \* userInput;

}

else

{

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

i--; // Retry the same index

}

}

// Find the largest element

int maxElement = myArray[0];

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

{

if (myArray[i] > maxElement)

{

maxElement = myArray[i];

}

}

// Print the array and the largest element

Console.WriteLine("\nArray:");

foreach (int value in myArray)

{

Console.Write(value + " ");

}

Console.WriteLine($"\nLargest element: {maxElement}");

Console.ReadLine();

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Console.WriteLine("FOR LOOP");

Console.WriteLine("Number | Square Root");

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

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

{

double sqrRoot = Math.Sqrt(i);

Console.WriteLine($" {i} = {sqrRoot:F4}");

}

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Console.ReadLine();

}

}

}

**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 FILE**

<!DOCTYPE html>

<html lang="en">

<head>

<title>Number Checker</title> <!-- title of the webpage -->

<style> <!-- CSS that targets the styling of the body of HTML -->

body

{

font-family: Calibri;

}

</style>

</head>

<body>

<br>

<h1 align = "center">Even or Odd number Checker</h1>

<input type="number" id="numberInput" placeholder="Enter a number"> <!-- for user input -->

<button onclick="checkEvenOrOdd()">Check</button> <!-- clickable button with onclick attribute that specifies Java function -->

<p id="result"></p> <!-- paragraph tag that with id attribute that assigns a unique identifier to this paragraph -->

<script src="script.js"></script> <!-- calls out the external file which is the javascript -->

</body>

</html>

* **Javascript File**

function checkEvenOrOdd() {

const inputNumber = parseInt(document.getElementById("numberInput").value); //Converts the input value from html id"number input" into integer.

if (isNaN(inputNumber)) {

document.getElementById("result").textContent = "Please enter a valid number."; //will execute if NaN returns true

} else {

const result = inputNumber % 2 === 0 ? "even" : "odd"; //checks if input value is divisible by 2

document.getElementById("result").textContent = `The number is ${result}.`; //will execute if input value id divisible by 2, result return as string

}

}

**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.

It’s an approach that focuses on defining and sculpting named classes as entities with attributes. OOP provides a structured approach to software development, making code more modular, maintainable, and adaptable. It makes reusing and maintaining code easier.

* Encapsulation: Encapsulates data and methods within a class, hiding the internal details from the outside world. This data hiding improves security and prevents unintended modifications.

Scenario: Building a library management system.

* Create an Item class with attributes like title, author, and availability. Encapsulate these attributes by making them private and provide public methods (getters and setters) to access and modify them. By encapsulating the data, you ensure that only authorized methods can change the availability status or retrieve book details.
* Abstraction – Focuses on essential features while hiding unnecessary complexities. It allows you to create high-level interfaces that simplify interactions with objects.

Scenario: Developing a payment gateway integration for an e-commerce platform.

* Create an abstract class PaymentGateway with methods like processPayment() and refund().

Concrete classes like PayPalGateway, StripeGateway, and CreditCardGateway extend PaymentGateway.

Each subclass implements its specific payment processing logic (e.g., API calls, encryption). Abstraction allows the e-commerce platform to interact with any payment gateway without worrying about implementation details.

* Inheritance – Enables you to create new classes based on existing ones. It promotes code reuse by allowing a new class (the subclass) to inherit properties and behaviors from an existing class (the super class)

Scenario: Designing a vehicle rental system.

* Create a base class Vehicle with common properties like make, model, and rentalPrice. Derived classes like Car, Motorcycle, and Van inherit from Vehicle. Each subclass adds specific attributes (e.g., numDoors, engineType) and methods (e.g., startEngine()). Inheritance ensures consistent behavior across different vehicle types while minimizing redundant code.
* Polymorphism – Allows objects of different classes to be treated uniformly. It enables flexibility by allowing a single interface to represent various implementations.

Scenario: Developing a shape drawing application.

* Define an abstract class Shape with methods like calculateArea() and draw(). Concrete classes like Circle, Rectangle, and Triangle implement these methods differently. Users can interact with any shape object using the same interface (e.g., calculateArea()), regardless of the specific shape. Polymorphism allows flexibility in handling various shapes without knowing their exact types.

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.