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.Write(“Enter the base of the triangle: ”);  
 double triangleBase = Convert.ToDouble(Console.ReadLine());  
  
 Console.Write(“Enter the height of the triangle: ”);  
 double triangleHeight =Convert.ToDouble(Console.ReadLine());  
  
 double area = 0.5 \* triangleBase \* triangleHeight;  
  
 Console.WriteLine(“The are of the triangle is: “ + area);  
 }  
}**

This program first prompts the user to enter the base and height of the tringle. It then calculates the area the formula 0.5 \* base \* height. Finally, it outputs the calculated area to the console.

**(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++)  
 {  
 Console.Write(“Enter a number: ”);  
 int number = Convert.ToInt32(Console.ReadLine());  
 array[i] = number \* number;  
 }  
  
 int max = array[0];  
 for (int i = 1; I < array.Length; i++)  
 {  
 if (array[i] > max)  
 {  
 max = array[i];  
 }  
 }  
 }  
}**

**This program first prompts the user to enter 5 numbers. Each number is squared and stored in the array. The program then iterates over the array to find the largest element, which is finally printed to the console.**

**(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 ()  
 {  
 for (int i = 1; I <= 10; i++)  
 {  
 double sqrt = Math.Sqrt(i);  
 Console.WriteLine(“The square root of ” + i + “ is “ + sqrt);  
 }  
 }  
}**

This program uses a for loop to iterate over the numbers 1 through 10. For each number, it calculates the square root using the Math.Sqrt method and then prints the number and its square root to the console.

**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.

**<!DOCTYPE html>  
<html>  
<head>  
 <title>My Website</title>  
 <link rel="stylesheet" type="text/css" href="styles.css">  
</head>**

**<body>  
 <h1>Welcome to...</h1>  
 <p>This is a paragraph...</p>  
 <img src="image.jpg" alt="Relevant image">  
 <ol>  
 <li>Ordered Item 1</li>  
 <li>Ordered Item 2</li>  
 <li>Ordered Item 3</li>  
 </ol>  
 <ul>  
 <li>Item 1</li>  
 <li>Item 2</li>  
 </ul>  
 <p><a href="https://www.example.com">Link to an external website</a></p>  
 <h3>This is a heading with inline CSS</h3>  
</body>  
</html>**

**body {  
 background-color: lightblue;  
}  
h1, h2, h3 {  
 padding: 20px;  
}  
p {  
 font-size: 14px;  
}  
li {  
 list-style-type: disc;  
}  
h3 {  
 color: blue;  
}**

In the HTML, I've added an image tag with a placeholder source, an ordered list with three items, a hyperlink within a paragraph tag, and an inline style to change the color of the h3 heading.

In the CSS, I've set the background color of the body to light blue, applied a padding of 20px to all headings, set the font size of the paragraph tag to 14px, and changed the list style of list items to disc (bullet points).  
  
  
**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>  
<body>  
 <input type="number" id="num" placeholder="Enter a number">  
 <button onclick="checkNumber()">Check Number</button>  
 <p id="result"></p>  
  
 <script src="script.js"></script>  
</body>  
</html>**

**function checkNumber() {  
var num = document.getElementById('num').value;  
var result = document.getElementById('result');**

**if (num % 2 == 0) {  
 result.textContent = num + " is even.";  
 } else {  
 result.textContent = num + " is odd.";  
 }  
}**

**In the HTML, I've added an input field for the number, a button that calls the checkNumber function when clicked, and a paragraph where the result will be displayed.**

**In the JavaScript, the checkNumber function gets the value of the input field and the paragraph. It then checks if the number is even or odd by using the modulus operator (%). If the remainder of the division by 2 is 0, the number is even; otherwise, it's odd. The result is then displayed in the paragraph.**

**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 (OOP) is a programming paradigm that uses "objects" – data structures consisting of data fields and methods together with their interactions – to design applications and computer programs. It is a fundamental approach to software development that greatly enhances the efficiency, scalability, and maintainability of code.  
  
**Encapsulation**: This principle is about hiding the internal details of how an object works. It allows us to bundle data (attributes) and the methods that operate on that data into a single unit called a class. An object is an instance of a class. Encapsulation provides a way to protect data from accidental corruption and misuse. For example, a class BankAccount might have private attributes for account number and balance, and public methods for deposit and withdrawal. The internal workings of these methods are hidden from outside classes.  
  
**Inheritance**: This principle allows one class to inherit properties and methods from another class, promoting code reusability. For example, a Vehicle class might have properties like make, model, and color, and methods like start and stop. A Car class could inherit from Vehicle, gaining all its properties and methods without having to define them itself. The Car class could then add more specific properties or methods, like convertible.  
  
**Polymorphism**: This principle allows one interface to be used for a general class of actions. It means that the correct function is called for an object, depending on the nature of the object. For example, you might have a Shape class with a method draw(), and subclasses Circle, Square, and Triangle. Each subclass would implement draw() differently, but you could call draw() on any array of shapes, and the correct method would be executed.  
  
**Abstraction**: This principle is about simplifying complex systems by modeling classes appropriate to the problem, and working at the most appropriate level of inheritance for a given aspect of the problem. For example, a Car class might have a start() method. This method is an abstraction of the process of starting a car—the details of how the engine, fuel system, and electrical system work together to achieve this are hidden from the user.  
  
  
  
  
  
  
Real-world scenarios where OOP is particularly valuable include:

**Game** **Development**: Each object in the game can be an instance of a class, with properties like size, position, and color, and methods to move, collide, etc. Different types of objects can inherit from a general class.

**GUI** **Applications**: Objects in these applications often correspond to physical objects on the screen, like buttons and sliders. These objects can inherit from general classes like Button or Slider.

**Web** **Development**: Backend frameworks like Django (Python) and Rails (Ruby) use OOP principles to model user data and the relationships between different data entities.

In conclusion, OOP principles are crucial in software development as they help to reduce complexity, increase reusability, and improve maintainability of code. They provide a clear structure for the programs, making them easier to design, develop, and manage.

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.