**BUCALOY, APRILROSE BSIT32E2**

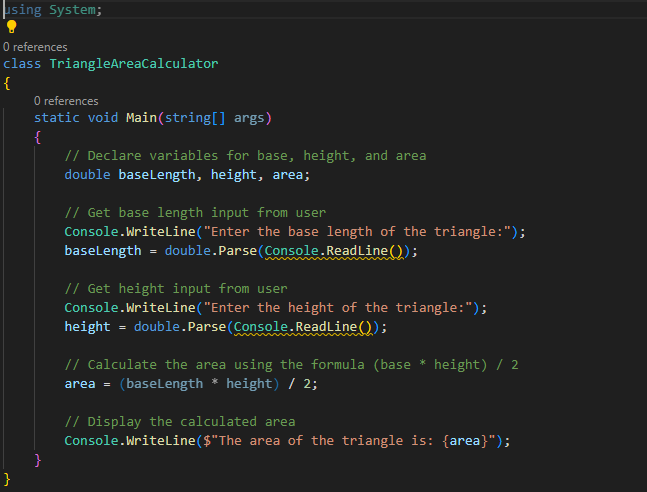
**SAMPLE ASSESMENT FOR INTRODUCTION TO PROGRAMMING**

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.

public class Triangle Area {

public static void Main(string[] args)

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

double base = Convert.ToDouble(Console.ReadLine());

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

double height = Convert.ToDouble(Console.ReadLine());

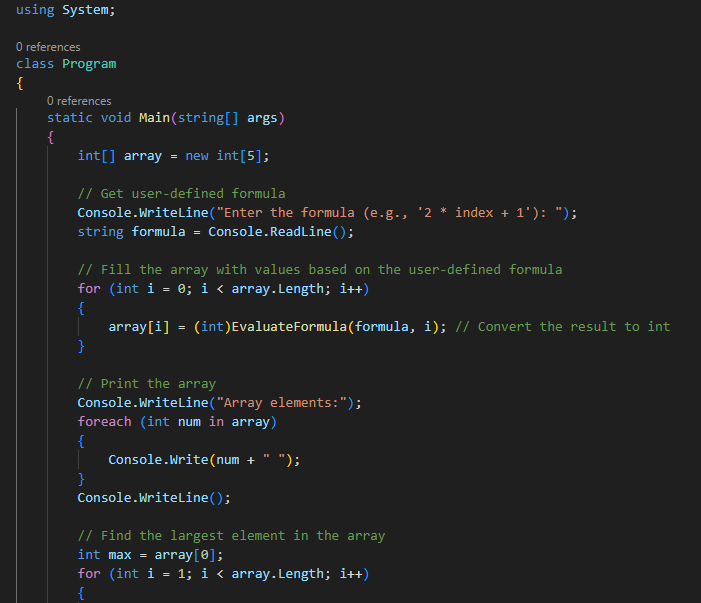
double area = 0.5 \* base \* height.

Console.WriteLine("The area of the triangle 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.**

****

int [] numbers = new int[5];

Console. Write("Enter formula for array values (use n for element number): ");

string formula = Console.ReadLine();

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

// Evaluate formula with current index

numbers[i] = (int) Math. Pow(i, 2);

}

int max = numbers[0];

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

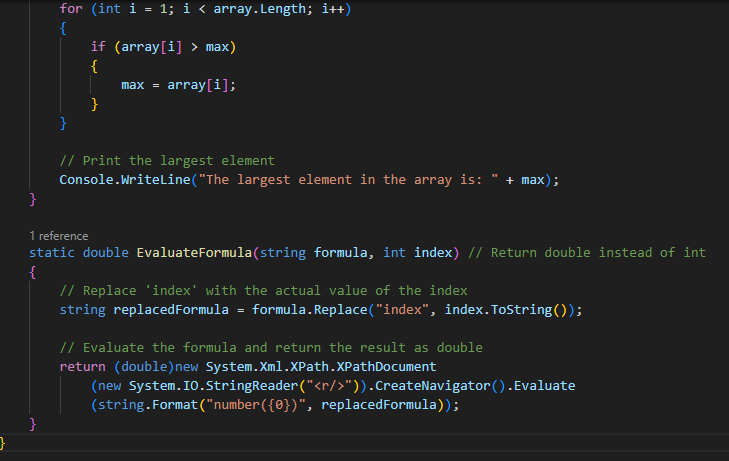
if(numbers[i] > max) {

max = numbers[i];

}

}

Console. WriteLine("The largest number is: " + max);

****

**(10 points) Implement a simple for loop that iterates from 1 to 10 and prints each number along with its square root.**

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

{

Console.WriteLine(i);

double squareRoot = Math.Sqrt(i);

Console.WriteLine(squareRoot);

}

Explanation:

for loop starts with defining an integer variable i and initializing it to 1

The second part i <= 10 is the loop condition that will continue running as long as i is less than or equal to 10

i++ increments i by 1 after each iteration

Inside the loop body:

Console.WriteLine(i); prints out the current value of i

Math.Sqrt(i) calculates the square root of the current i value

The square root result is stored in square root

Console.WriteLine(squareRoot); prints out the square root

So on each iteration, it will print the number, calculate and print its square root, then increment i and repeat until i becomes 11. This prints the numbers from 1 to 10 along with their square roots.

**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>**

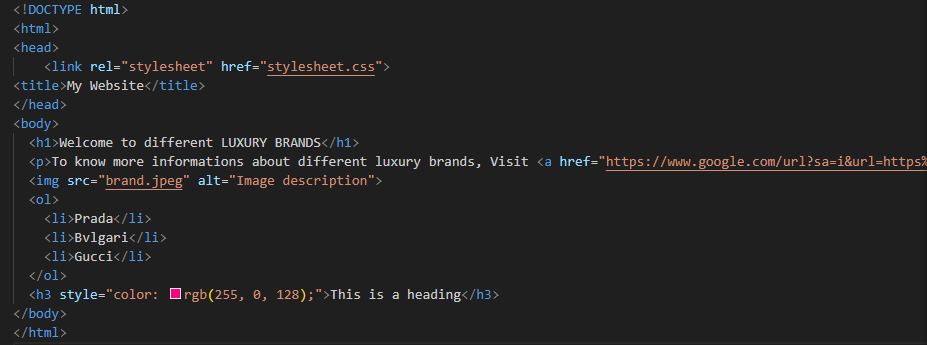
**</html>**

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.

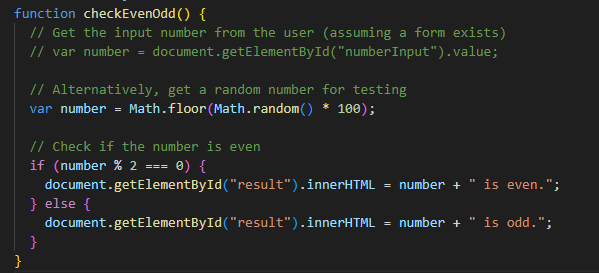
Apply a padding of 20px to all headings (h1, h2, h3).

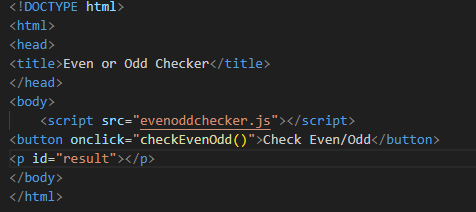
Set the font size of the <p> tag to 14px.

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.

****

****

**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: The use of classes, objects, and well-defined interfaces facilitates teamwork, as different developers can work on different components independently while ensuring seamless integration. Object-oriented programming refers to the concept in high-level languages such as Java and Python that uses Objects and classes in their implementations. OOP has four major building blocks which are Polymorphism, Encapsulation, Abstraction, and Inheritance.  There are other programming paradigms such as Procedural programming in which codes are written in sequentially.

The public methods that we decide to expose are all that are needed for users of the parent class to communicate with complex subclass implementations. Again, this promotes reuse and modular programming.

**4 KEY PRINCIPLES OF OOP**

**ENCAPSULATION** - This is a programming style where implementation details are hidden. It reduces software development complexity greatly. With Encapsulation, only methods are exposed. The programmer does not have to worry about implementation details but is only concerned with the operations.

**INHERITANCE** - Objects can interact with one another by using the properties of each block or extending the functionalities of a block through inheritance.  Inheritance ensures that codes are reused. There are millions of Java and Python libraries that a programmer can use through inheritance.

**POLYMORPHISM -** Polymorphism is a popular concept in object-oriented programming ([OOP](https://www.techtarget.com/searchapparchitecture/definition/object-oriented-programming-OOP)), referring to the idea that an entity in code such as a variable, function or [object](https://www.techtarget.com/searchapparchitecture/definition/object) can have more than one form. The word *polymorphism* is derived from Greek and means "having multiple forms." Apart from computer programming, the idea of polymorphism occurs in other real-world areas, including biology, chemistry, and drug development.

**ABSTRACTION -** Abstraction is an essential feature of Object-Oriented Programming, which implies only keeping the essentials and removing unnecessary information. Abstraction in OOPS is used to hide unnecessary information and display only necessary information to the users interacting. It is essential to represent real-world objects in a simplified manner for users to interact easily.

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.