**Name:** Llantos, Marygrace R.

**Course/Section:** BSIT 32E3

**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.

**Code:**

using System;

class Program

{

static void Main()

{

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

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

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

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

double area = 0.5 \* baseLength \* 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.**

**Code:**

using System;

namespace ArrayExample

{

class Program

{

static void Main(string[] args)

{

// Create an array of 5 integers

int[] array = new int[5];

// Fill the array with values based on a user-defined formula (n^2)

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

{

Console.Write("Enter a number: ");

int n = Convert.ToInt32(Console.ReadLine());

array[i] = n \* n;

}

// Find and print the largest element in the array

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.**

**Code:**

using System;

namespace SimpleForLoop

{

class Program

{

static void Main(string[] args)

{

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

{

double squareRoot = Math.Sqrt(i);

Console.WriteLine("Number: " + i + " Square Root: " + 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.

**Code:**

<!DOCTYPE html>

<html>

<head>

<title>My Website</title>

</head>

<body>

<h1>Welcome to My Website</h1>

<p>This is a paragraph with a <a href="https://www.example.com" target="\_blank">link to an external website</a>.</p>

<ul>

<li>Item 1</li>

<li>Item 2</li>

<li>Item 3</li>

</ul>

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

<img src="image.jpg" alt="An image on my website">

</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.

**Code:**

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.

**Code:**

**HTML**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Even or Odd Checker</title>

</head>

<body>

<button id="check-button">Check Even/Odd</button>

<p id="result"></p>

<script src="script.js"></script>

</body>

</html>

**JAVA**

const checkButton = document.getElementById('check-button');

const resultParagraph = document.getElementById('result');

function isEven(number) {

return number % 2 === 0;

}

checkButton.addEventListener('click', function() {

const inputNumber = Number(prompt('Enter a number:'));

if (isNaN(inputNumber)) {

resultParagraph.textContent = 'Please enter a valid number.';

} else {

const evenOdd = isEven(inputNumber) ? 'even' : 'odd';

resultParagraph.textContent = `The number ${inputNumber} is ${evenOdd}.`;

}

});

**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 effectiveness of object-oriented programming (OOP) in improving code organization, maintainability, and scalability makes it the basis of contemporary software development. Code security and adaptability are emphasized by encapsulation, the idea of grouping variables and methods inside a class to prevent direct access from third-party sources. To strengthen data integrity and confidentiality, for example, in a banking system, private account information can be contained inside a class to prevent unwanted access.

Code reuse is made possible by inheritance, which encourages a hierarchical structure by making it easier to create new classes based on existing ones. Flexibility and extensibility are encouraged by polymorphism, which enables objects to be treated similarly despite their various implementations. In a vehicle management system, for example, a generic class called "Vehicle" may behave polymorphically to support a variety of vehicle kinds, including cars, trucks, and motorbikes. By hiding pointless features, abstraction helps to simplify complicated systems and fosters a better comprehension of their functionality. Coding is more transparent and efficient in healthcare applications when standard features are reduced under a "Patient" class. Developers can create software that is naturally responsive to changing needs, is more efficient, and is easier to maintain by implementing these OOP concepts.