# GERNALE, ALBERT D. BSIT22A3

**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;*

*namespace TriangleArea*

*{*

*class Program*

*{*

*static void Main(string[] args)*

*{*

*// Request user input for the base and height of the triangle*

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

*string baseInput = Console.ReadLine();*

*float baseValue = float.Parse(baseInput);*

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

*string heightInput = Console.ReadLine();*

*float heightValue = float.Parse(heightInput);*

*// Calculate the area of the triangle*

*float area = (baseValue \* heightValue) / 2;*

*// Display the calculated area*

*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.**

*using System;*

*class Program*

*{*

*static void Main()*

*{*

*// Declare an array of 5 integers*

*int[] numbers = new int[5];*

*// Fill the array with values based on a user-defined formula*

*Console.WriteLine("Enter 5 integers:");*

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

*{*

*Console.Write("Enter number " + (i+1) + ": ");*

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

*numbers[i] = n \* n;*

*}*

*// Print the largest element in the array*

*int max = numbers[0];*

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

*{*

*if (max < numbers[i])*

*{*

*max = numbers[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.**

*using System;*

*class Program*

*{*

*static void Main()*

*{*

*// Iterate from 1 to 10 and print each number along with its square root*

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

*{*

*double sqrt = Math.Sqrt(i);*

*Console.WriteLine("The square root of " + i + " is " + sqrt);*

*}*

*}*

*}*

**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.

*<!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>*

*<h1>Even or Odd Checker</h1>*

*<input type="number" id="numberInput" placeholder="Enter a number">*

*<button onclick="checkEvenOdd()">Check</button>*

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

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

*</html>*

*function checkEvenOdd() {*

*const number = document.getElementById("numberInput").value;*

*// Convert the input value to a number (in case it's a string)*

*const numberAsNumber = parseInt(number);*

*if (numberAsNumber % 2 === 0) {*

*document.getElementById("result").textContent = "The number is even.";*

*} else {*

*document.getElementById("result").textContent = "The number is odd.";*

*}*

*}*

**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 like a toolbox for building software. It helps developers organize their code into easy-to-handle parts called objects, which have their own data and actions they can perform. OOP has four main ideas:*

* *Encapsulation: Think of encapsulation as wrapping a gift. It keeps everything neatly packaged so that people outside can't mess with what's inside unless they use the right methods.*
* *Inheritance: Inheritance is like passing down traits from parents to children. It allows new objects to inherit qualities and actions from older ones, helping to reuse code and make it more organized.*
* *Polymorphism: Polymorphism is like a shape-shifting superhero. It lets objects of different types be treated as if they were all the same type, making it easier to write flexible and adaptable code.*
* *Abstraction: Abstraction is like looking at a painting from far away. You can see the main picture without getting bogged down in the tiny details. Similarly, abstraction in programming helps focus on the important parts while hiding the complex inner workings.*

*These principles are useful in many situations:*

* *Big software projects: OOP helps keep large projects organized and easier to manage.*
* *Making software tools: OOP makes it easier to create reusable pieces of code that can be used in different programs.*
* *Building interfaces: When making things like apps or websites, OOP helps represent elements like buttons and menus as objects, making them easier to work with.*
* *Creating games: OOP is great for building video games because it lets developers model game characters, environments, and actions in a structured way.*