Aerica Mae Arandia

BSIT32E1

IT ELECTIVE 3

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.

Answer:

using System;

class Program

{

static void Main(string[] args)

{

double BLength, Height, Area;

// Base

Console.Write("Enter the Base of the Triangle: ");

while (!double.TryParse(Console.ReadLine(), out BLength) || BLength <= 0)

{

Console.WriteLine("Please enter a valid positive number for the Base.");

Console.Write("Enter the Base of the Triangle: ");

}

// Height

Console.Write("Enter the Height of the Triangle: ");

while (!double.TryParse(Console.ReadLine(), out Height) || Height <= 0)

{

Console.WriteLine("Please enter a valid positive number for the Height.");

Console.Write("Enter the Height of the Triangle: ");

}

//Area

Area = 0.5 \* BLength \* Height;

//result

Console.WriteLine($"The Area of the Triangle with Base {BLength} and Height {Height} is: {Area}");

Console.WriteLine("\nEnter 5 Integers for the array:");

// array of 5 Integers

int[] arr = new int[5];

// enter the array

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

{

Console.Write($"Enter value {i + 1}: ");

if (!int.TryParse(Console.ReadLine(), out arr[i]))

{

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

i--; // Decrement i to retry entering the current element

}

}

// Print the array elements

Console.WriteLine("\nArray elements:");

foreach (int num in arr)

{

Console.Write(num + " ");

}

Console.WriteLine();

// Find the largest element in the array

int max = arr[0];

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

{

if (arr[i] > max)

{

max = arr[i];

}

}

// Print the largest element

Console.WriteLine("\nLargest element in the array: " + max);

// Print the square root of numbers from 1 to 10

Console.WriteLine("\nSquare roots of numbers from 1 to 10:");

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

{

double squareRoot = Math.Sqrt(i);

Console.WriteLine($"Number: {i}, Square Root: {squareRoot}");

}

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, javascript, css code:

<!DOCTYPE html>

<html>

<head>

<title>My Website</title>

<link rel="stylesheet" type="text/css" href="arandia.css">

</head>

<body>

<h1>Welcome to...</h1>

<p>This is a paragraph...</p>

<img src="ae.jpg" alt="Description of the image" />

<ul>

<li>Item 1</li>

<li>Item 2</li>

</ul>

<ol>

<li>First Item</li>

<li>Second Item</li>

<li>Third Item</li>

</ol>

<p>

This is a <a href="https://github.com/Arandia-AericaMae-BSIT32E1">link</a> to an external website.

</p>

<h3 style="color: red">Heading 3</h3>

<button onclick="checkNumber()">Check Number</button>

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

<!-- JavaScript-->

<script>

function checkNumber() {

var number = parseInt(prompt("Enter a number:"));

if (isNaN(number)) {

document.getElementById("result").textContent =

"Please enter a valid number.";

} else {

if (number % 2 === 0) {

document.getElementById("result").textContent =

number + " is even.";

} else {

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

}

}

}

</script>

</body>

</html>

body {

background-color: lightblue;

}

h1,

h2,

h3 {

padding: 2.5vh;

}

p {

font-size: 1.75vh;

}

li {

list-style-type: disc;

}

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.

The capacity of object-oriented programming (OOP) to improve code organization, reusability, and maintainability makes it essential to software development. Encapsulation, inheritance, polymorphism, and abstraction are the guiding principles that greatly contribute to these advantages.

In encapsulation, Data and methods that work with the data are bundled together into a single unit or class through the process of encapsulation. This facilitates a distinct division between the implementation and the interface by limiting access to an object’s internal details. For example, encapsulation in a classroom scenario helps protect sensitive student data and ensures that any modifications to academic records are done in a controlled manner. Next is inheritance. enables classes to inherit traits and properties, which facilitates code reuse. A general "Shape" class in a graphic design application can be inherited by more specialized shapes, such as "Circle" or "Rectangle.". Next is polymorphism. Makes it easier to handle different things consistently. Various payment classes can implement a “Payment” interface in payment processing, increasing flexibility. Lastly is abstraction. Makes complex systems simpler by creating classes based on necessary characteristics. Common functionalities in a vehicle management system are represented by an abstract “Engine” class.

Developing graphical user interfaces (GUIs), managing large-scale systems, and building reusable software frameworks. Software development is made more effective, maintainable, and flexible by the combined application of OOP principles.

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.