CRISPULO CASTELLANES

BSIT – 32E2

Instructions: Read each question carefully and provide complete and clear answers. Avoid multiplechoice format responses. Focus on demonstrating your understanding through code, explanations, and discussions

Part 1.

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 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.

using System;

class Program

{

static void Main()

{

int[] numbers = new int[5];

// Use the length property of the array to determine its size

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

{

numbers[i] = (i + 1) \* (i + 1);

}

Console.WriteLine("Array elements:");

DisplayArrayElements(numbers); // Use a separate method for displaying array elements

int largestElement = FindLargestElement(numbers);

Console.WriteLine($"The largest element in the array is: {largestElement}");

}

static void DisplayArrayElements(int[] array)

{

foreach (var num in array)

{

Console.Write(num + " ");

}

Console.WriteLine();

}

static int FindLargestElement(int[] array)

{

if (array.Length == 0)

{

throw new ArgumentException("Array is empty");

}

int max = array[0];

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

{

if (array[i] > max)

{

max = array[i];

}

}

return 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()

{

For (int I = 1; I <= 10; i++)

{

Double squareRoot = Math.Sqrt(i);

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

}

}

}

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.

function checkEvenOrOdd(number) {

return number % 2 === 0 ? "Even" : "Odd";

}

<!DOCTYPE html>

<html>

<head>

<title>Even or Odd Checker</title>

<script>

Function checkAndDisplay() {

// Get the input value from the user or any other source

Var inputNumber = document.getElementById(“inputNumber”).value;

// Call the function and get the result

Var result = checkEvenOrOdd(inputNumber);

// Display the result in the paragraph element

Document.getElementById(“resultParagraph”).innerText = “The number is “ + result + “.”;

}

</script>

</head>

<body>

<h1>Even or Odd Checker</h1>

<label for=”inputNumber”>Enter a number:</label>

<input type=”number” id=”inputNumber”>

<button onclick=”checkAndDisplay()”>Check</button>

<p id=”resultParagraph”></p>

</body>

</html>

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 coding paradigm that organizes code effectively through the use of objects, instances of classes. The core principles of OOP namely encapsulation, inheritance, polymorphism, and abstraction, form the foundation for structuring and managing software systems. Encapsulation involves bundling data and methods into a class, ensuring clear interfaces such as in a Car class with encapsulated properties like color and methods like start and stop Inheritance allows classes to inherit properties and methods from others, promoting code reuse, as seen in a more specific Car class inheriting from a general Vehicle class. Polymorphism enables the treatment of different objects as a common type, simplifying code with a Shape class implementing calculateArea differently for specific shapes like Ciircle and Rectangle Abstraction conceals complex details, exemplified by a TV remote’s abstract controls enabling interaction without the need to understand internal workings. OOP’s significance lies in facilitating code reuse, enhancing modularity, scalability, and fostering collaboration among developers, particularly valuable in complex software systems such as GUI development, game development, and database systems. In summary, OOP provides a robust and adaptable approach to software development, emphasizing code organization, maintainability, and reusability, especially in intricate and structured systems.