Kaye Darleene A. Gatmaitan

BSIT 32E1

**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 BaseLength, Height, Area;

//Base

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

while (!double.TryParse(Console.ReadLine(), out BaseLength) || BaseLength <= 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 \* BaseLength \* Height;

//result

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

//enter values for the array elements

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

//enter 5 Integers

int[] arr = new int[5];

// show values

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();

// 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:

ANSWER:

HTML

<!DOCTYPE html>

<html>

<head>

<title>My Website</title>

</head>

<body>

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

<p><a href="https://www.example.com">Example Website</a>This is a paragraph...</p>

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

<ul>

<li>Item 1</li>

<li>Item 2</li>

<li>Item 3</li>

</ul>

</body>

</html>

CSS STYLE:

style

body {

background-color: lightblue;

}

h1, h2, h3 {

padding: 20px;

}

p{

font-size: 14px;

}

ul{

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.

ANSWER:

<-CSS STYLE->

body {

background-color: lightblue;

}

h1,

h2,

h3 {

padding: 20px;

}

p {

font-size: 14px;

}

li {

list-style-type: disc;

}

<-HTML->

<!DOCTYPE html>

<html>

<head>

<title>My Website</title>

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

</head>

<body>

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

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

<img src="image.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/Gatmaitan-KayeDarleene-BSIT32E1">link</a> to an external website.

</p>

<h3 style="color: blue">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>

**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:

OOP offers a standardized method for developing software, which encourages developer collaboration. The utilization of classes, objects, and clearly specified interfaces makes it easier for developers to collaborate since they may each focus on a separate component while ensuring an effortless integration.

**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.

**Example**

* If a developer wants to use a dynamic link library to display date and time, he does not have to worry about the codes in the date and time class rather he would simply use the data and time class by using public variables to call it up.

**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. The properties of a class can be inherited and extended by other classes or functions.

**Example**

* A real-world example of inheritance is a mother and child. The child may inherit attributes such as height, Voice patters, color. The mother can reproduce other children with the same attributes as well.

**Polymorphism**

* Polymorphism means existing in many forms. Variables, functions, and objects can exist in multiple forms in Java and Python. There are two types of polymorphism which are run time polymorphism and compile-time polymorphism. Run time can take a different form while the application is running and compile-time can take a different form during compilation.

**Example**

* A cursor may take different forms like an arrow, a line, cross, or other shapes depending on the behavior of the user or the program mode. With polymorphism, a method or subclass can define its behaviors and attributes while retaining some of the functionality of its parent class.

**Abstraction**

* Abstraction is concerned with ideas rather than events. It’s like a user running a program (Web Browser) without seeing the background codes. Abstraction is achieved in either Abstract classes or interface in Java and Python. NetBeans and Eclipse IDE implements abstraction for Java while Django implements abstraction for Python.

**Example**

* A programmer uses an Integrated Development environment to design a UI without worrying about how the IDE generates the HTML codes. In essence, abstraction displays the essential details for the user alone.