Test sample

1. What is the result of the following code snippet?

|  |
| --- |
| static void Main(string[] args){ int a = 10; int b = 20; Console.WriteLine(Math\_operation(a, b)); Console.ReadLine();}static int Math\_operation(int a, int b){ int result = (a + b) % b; return result;} |

* 15
* 10
* 23
* 11
1. What is the result of the following code snippet?

|  |
| --- |
|  static void Main(string[] args) { float [] arr = new float[] { 10.7f, 20.8f, 45.2f, 60.0f }; float sum = 0; foreach(float i in arr) { if (i < 30) sum += i; else sum += i / 2; } Console.WriteLine("Sum = " + sum); Console.ReadLine(); } |

* 87
* 78.5
* 84.1
* 87.4
1. What will be the output of the following code snippet?

|  |
| --- |
| class Box { public double length; // Length of a box public double breadth; // Breadth of a box public double height; // Height of a box } class Boxtester { static void Main(string[] args) { Box Box1 = new Box(); // Declare Box1 of type Box Box Box2 = new Box(); // Declare Box2 of type Box double volume = 0.0; // Store the volume of a box here // box 1 specification Box1.height = 5.0; Box1.length = 6.0; Box1.breadth = 7.0; // box 2 specification Box2.height = 10.0; Box2.length = 12.0; Box2.breadth = 13.0;  // volume of box 1 volume = Box1.height \* Box1.length \* Box1.breadth; Console.WriteLine("Volume of Box1 : {0}", volume); // volume of box 2 volume = Box2.height \* Box2.length \* Box2.breadth; Console.WriteLine("Volume of Box2 : {0}", volume); Console.ReadKey(); } } |

* 210 1500
* 1500 210
* 210 1560
* 1560 210
1. What will be the output of the following code snippet?

|  |
| --- |
|  class Program { public static void Main() { int[] arr = new int[5] { 10, 12, 6, 8, 1 }; int sum = 0; for (int i = 0; i < arr.Length; i++) { if (i % 2 == 0) sum += arr[i]; } Console.WriteLine("Sum = " + sum); } } |

* 22
* 17
* 16
* 21
1. What will be the output of the following code snippet?

|  |
| --- |
|  class Program { public static void Main() { int[] arr = new int[] { 6, 3, 9, 10, 1, 7, 12 }; int sum = 0; for (int i = 0; i < arr.Length; i++) { if (i % 3 == 0) sum += arr[i]; else sum += 1; } Console.WriteLine("Sum = " + sum); } } |

* 32
* 20
* 22
* 31

**Correspondence questions**

1. **Match terms and definitions correctly**

Object – Refers to a particular instance of a class where the object can be a

combination of variables, functions, and data structures

Method – A combination of instructions grouped together to achieve some result. It may take arguments and return result

Property– A member that provides a flexible mechanism to read, write, or compute the value of a private field

Field – a variable of any type that is declared directly in a class or struct.

1. **Match terms and definitions correctly**

Constructor– a public method with the same name as the class with no return type, which is called once upon object creation

Destructor **–** a special member function that is called when the lifetime of an object ends.

Overriding – allows you to change the functionality of a method in a child class

Overloading – was developed to reduce the number of different method names to be created by the programmer and it also makes life easier for the end object user

1. **Match terms and definitions correctly**

Array – a data structure that contains several variables of the same type

ArrayList – a special array that provides us with some functionality over and above that of the standard Array

Queue – a collection that is represented in the form of FIFO (First-In-First-Out) order

Stack – a collection of objects which is represented by LIFO

1. **Match terms and definitions correctly**

Abstraction – a process of showing only essential features of an object to the outside world and hide the other irrelevant information

Encapsulation – a process of binding data members (variables, properties) and methods together

Polymorphism – a process in which an object or function take different forms

Inheritance – a process of creating the new class by extending the existing class or the process of inheriting the features of base class

1. **Match terms and definitions correctly**

Protected – access is limited to the containing class or types derived from the containing class

Private – access is limited to the containing type

Public – access is not restricted

1. **Match terms and definitions correctly**

This – keyword is used to refer to instance members of the current class from within an instance method or a constructor

Virtual – keyword is used to modify a method, property, indexer, or event declaration and allow for it to be overridden in a derived class

Static – keyword is required to be used to address a class by its name and not its object

Sealed – keyword is used to restrict the inheritance feature of object-oriented programming

1. **Match terms and definitions correctly**

Checkbox – a control that allows the user to make single or multiple selections from a list of options

Messagebox – an interface between the user and the application

Radiobutton – a control that allows the user to make multiple selections from a list of options in the same group

ColorDialog – allows to choose a color from a palette

1. **Match terms and definitions correctly**

Partial – a feature which allows us to write class across multiple files

Enum – a value type defined by a set of named constants of the underlying integral numeric type

Void – the return type of a method (or a local function) to specify that the method doesn't return a value

1. **Match terms and definitions correctly**

Windows forms – an application with graphical user interface

Button – an element that is usually used to be clicked on

Property window – a window where different parameters of an element are changed

1. **Match terms and definitions correctly**

FileStream – Provides a Stream for a file, supporting both synchronous and asynchronous read and write operations

StreamReader – Implements a TextReader that reads characters from a byte stream in a particular encoding

IOException – The exception that is thrown when an I/O error occurs