**Section A Theory Questions**

|  |
| --- |
| **115363: Apply fundamental principles of Object-Oriented Programming to solve a given problem** |

1. Identify the following from the code given below (5)(SO1:AC1,2,3)
2. **Method used**   
    Main method
3. **String variable** str
4. **The class name** Test
5. **The namespace used** System

using System; class Test

{

public static void Main(string[] args)

{

int numb = 0; string str = "Hey";

}

}

1. Use the code below to answer the following questions: (5) (S03:AC6)

25 }

}

int addValues(int val, int addVal)

{

return val + addVal;

}

class Two : One

{

public static void Main(string[] args

{

Two two = new Two();

int i = two.changeValue(10); int ans = two.addValues(i, 5);

Console.WriteLine("The answer is: " + ans);

}

public class One

{

public int changeValue(int val)

{

return val;

}

using System;

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

1. What does Line 14 do?  
    Instantiate an object
2. Briefly explain Line 10   
     
   class Two inherits the base class(Class One)
3. Explain what happens at Line 22  
     
   an integer value returns the sum of val + addVal
4. State the use of Line 16   
     
   calls another method named addValues on the two object, passing the values of i and 5 as arguments. The return value is stored in the variable ans.
5. State the main function or purpose of the following with reference to the code below. (5)( SO2:AC1)
6. **#define** - Defines a constant or a macro for text substitution. #define TESTING defines a constant named TESTING
7. **Using System** -Imports the System namespace, which provides access to a wide range of commonly used classes, methods, and types in C#. In the code: using System; allows the code to use features like Console for writing to the console without having to repeatedly specify the full namespace path
8. **#if…#else…#endif** - Implements conditional compilation, enabling different code blocks to be included or excluded based on whether certain conditions are met. **#if TEST**: Checks if the symbol TEST is defined. If it is, the code within this block will be included. **#else**: Provides an alternative code block to be executed if the #if condition is not met. **#endif**: Marks the end of the conditional compilation block.
9. **Main() -** Represents the entry point of a C# program, where program execution begins. public static void Main(string[] args) defines the Main method, which is the first method to be called when the program runs.

#define TESTING using System;

namespace Tester

{

public class MainClass

{

public static void Main(string[] args)

{

#if TEST

Console.WriteLine("Testing"); #else

Console.WriteLine("Not testing"); #endif

Console.ReadLine();

}

}

}

1. Look closely at the program given below. The code is incomplete and has been replaced by bold letters in brackets. Fill in the gaps. (5) (SO2:AC1)

using System;

class Test

{

public **Static** void Main(string[] args)

{

int **i**, j, k;

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

{

Console.WriteLine("Loop 1: {0}", i); if (i == 2)

{

for (**j = 0**; j < 3; j++) // this loops 3 times

{

Console.WriteLine("Loop 2: {0}", j);

if (j **==** 1) // checks if the values are equal

{

for (k = 0; k < 2; k++)

{

Console.WriteLine("Loop 3: {0}", k);

}

}

}

}

}

}

}

1. Referring to the code block below. Determine the: (5) (SO2:AC1)
2. **First line of output** - "Flower 1: daffodil”
3. **Array declaration and initialization** - string[] flowers = { "daffodil", "rose", "tulip" };
4. **Statement used to loop through a certain collection** - foreach (string flower in flowers)
5. **Statement decrementing int I** – i-=1
6. **Class** - Test

using System;

class Test

{

public static void Main(string[] args)

{

string[] flowers = { "daffodil", "rose", "tulip" };

foreach (string flower in flowers)

{

Console.WriteLine("Flower 1: " + flower);

}

for (int i = flowers.Length - 1; i >= 0; i--)

{

Console.WriteLine("Flower2: " + flowers[i]);

}

}

}

|  |
| --- |
| **115378: Demonstrate an understanding of advanced object-oriented programming** |

1. State the following by making reference to the code below. (5) (S01:AC1,2)
2. **Base Class** – Animal
3. **Derived Class** – Dog
4. **New Instance** – a = new Animal();
5. **Entry point of the program** - Main

public class Animal

{

public void MakeSound();

}

public class Dog : Animal

{

public static void Main()

{

Animal a = new Animal(); a.MakeSound();

}

}

7. Explain what the following lines of code do referring to the given code snippet below. (5) **Line 1** – Declaring a variable named age of type integer and initialises with value 20

**Line 4** – This line starts an if statement that checks if either the person's age is greater than 15 or their gender is female (using the XOR operator). If either condition is True, the if block will be executed.

**Line 14** – Prints out to the word “ Childish “ to the console.

**Line 8** - This line starts an elif statement that checks if both the person's age is greater than 21 and their gender is female (using the AND operator). If both conditions are True, the elif block will be executed.

**Line 12** - This line starts an else block that will be executed if neither of the if or elif conditions are met.

12 else 13 {

14 Console.WriteLine("Childish!!"); 15 }

Console.WriteLine("All grown up!");

if ((age > 15) ^ (gender == 'f'))

{

Console.WriteLine("Growing up...");

}

else if ((age > 21) && (gender == 'f'))

{

10

11 }

int age = 20;

char gender = 'm';

1

2

3

4

5

6

7

8

9

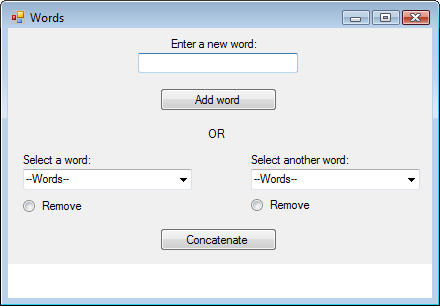
1. State the most accurate word for the given descriptions. (5) (SO2:AC1)
2. Loop that executes at least once and then executes a group of statements continually while the condition is true.  
   - do while
3. Loop used when you know how many times it must be executed.  
   -for loop
4. Loop that is designed to execute as long as the condition being checked remains true.  
   - while loop
5. Statement used when you want to skip the rest of the current execution and begin the next one immediately.  
   -break
6. Used to store a single character  
   -char
7. State the appropriate operators which symbolizes the given statements below. (5) (SO2:AC1)
8. Evaluates to true if the first value is equal to the second value, false otherwise.  
   **==**
9. Evaluates to true if the first value is not equal to the second value, false otherwise.  
   **!=**
10. Evaluates to true if the first value is greater than or equal to the second value, false otherwise.  
    **>=**
11. Evaluates to true if the first value is smaller but not equal to the second value, false otherwise.  
    **<**
12. Condition on the left is checked and if it equals to false, then the whole statement equals false without checking the condition on the right.  
    **&&**
13. Give the operator name or the operator sign as best described by the following descriptions. (5) (SO2:AC1)
14. Changes the sign of the value.  
    unary minus (-)
15. Decreases the value by one.  
    decrement operator (--)
16. Inverts the value of the Boolean variable of the expression.  
    logical Not operator (!)
17. Has no effect.  
    identity operator (=)
18. Subtracts the right operand from the left operand and returns the value  
    subtraction operator (-)

**Section B: Practical question**

|  |
| --- |
| **115381: Apply the principles of creating a computer program using an OOP language in a GUI environment** |

1. Create a program that allows the user to enter a new word or concatenate two words. Display the concatenated word in the Label control.(S01:AC1, SO3:AC1|)

The application should look as follows:



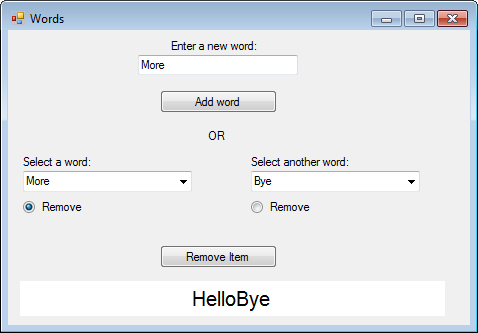
Combo boxes

Label

The following checks must be in place if the user chooses to enter a new word:

* + Check that there is a word entered into the TextBox control when the user presses the Add button.
  + Check that the word has not been entered more than once.
  + Check if the TextBox control is empty or contains the same word twice. If so, display a MessageBox with an appropriate message.
  + Display a MessageBox to inform the user that the new word has been added.

Alternatively, the user can select or remove a word from both the ComboBox controls before pressing the concatenate button.



The following must be in place for removing a word from a combo box:

* + Ensure that a word has been selected from a ComboBox control.
  + If not, display a MessageBox with an appropriate message.
  + When the **Remove Item** radio button is clicked, the concatenate button text must read ‘Remove Item’.
  + Check which radio button is selected and remove the word from the corresponding ComboBox when the Remove Item button is clicked.
  + Display a MessageBox with an appropriate message when a word is removed.

Ensure that the application uses only one button for removing and one button for concatenating the words. The following checks must be in place for concatenating two words:

* + Ensure that a word has been selected from both ComboBox controls.
  + If not, display a MessageBox with an appropriate message.
  + Check that different words have been selected from the ComboBox controls, i.e. ComboBox one cannot have the same word selected as ComboBox two.

The new words added must be placed in an ArrayList collection when the Add button is pressed. Another ArrayList collection must be created to store the words after these have been concatenated. Display a MessageBox notifying the user that the word has been added to the collection.

The Label control’s background colour must be set to white and the font size of the Label control must be set to 10. Set the Dock property of the Label control to bottom. Once the concatenated word is then displayed in the label, the application should look as follows:

