



Course number : 420-CT2-AS

OBJECT ORIENTED PROGRAMMING

Teacher : Maftai Mihai

Weighting :	30%	Group :	07264
Points number:	100	Date :	19-06-2018
Duration :	4 periods of 50 min	Session :	Summer 2019

STATEMENT OF THE COMPETENCY

- To use object-oriented development approach (016T)

REQUIRED COMPETENCIES FOR THE EXAMS

1. To create an object model
2. Refine the object model.
3. To program a class
4. To ensure that the class functions correctly

DIRECTIVES

- Open book and notes.

INSTRUCTIONS

This exam has 1 section of 10points and 3 sections of 30 points.

The exam has 4 sections:

(Theory, Design, & Coding for a Form and console application in C#), evaluated like this:

Theory	Presentation level	Business level	Console OOP app.	Total
10 points	30 points	30 points	30 points	100 p

Section 1 – 10 points

Q1 - Which call to the method that follows, is a correct one? Choose all that applies:

```
public byte Add(byte nbr1, byte nbr2) { ... }
```

- a. `Console.WriteLine("{1} + {2} = {0}", Add(50, 70), 70, 50);`
- b. `Console.WriteLine("The total is :" + Add(155, 145));`
- c. `Console.WriteLine("The total is :" + Add(Add(10, 40), Add(40, 30)));`
- d. `Console.WriteLine("The total is :" + Add(300, 70));`

Q2 - We have: `string s1= "Bonjour", s2;` and the instruction: `s2 += s1;` choose all that are true:

- a. `s1.Length == s2.Length`
- b. `s1.Length + s2.Length = 16`
- c. `s2.Length = 7`
- d. `s2.Length = 8`

Q3 What happens when this statement is executed? Choose all that are true.

```
Auto car = new Auto("Ford");
```

- a. An object Ford of the Auto class is created.
- b. An object car of the Auto class is declared, and created.
- c. A constructor that accepts a string parameter of Auto class is called.
- d. The default constructor of the Auto class is called.

Q4 - Which method is an example of overloading the method that follows? Choose all that are true.

```
public int Mul(int num) { ... }
```

- a. `public int Mul(byte num, int num2) { ... }`
- b. `public int Mul(int num, int num2) { ... }`
- c. `public int Mul(int num, byte num2) { ... }`

Q5 The declaration of a property, can be done before you declare the variables of the class:

- a. Yes, but you have to declare the property as private
- b. Yes, only if you have public variables
- c. No, the order is important
- d. Yes, the order is not important

Your answers will be presented into the form application when the **Show Answers** button is clicked.
Write **none** into the read-only textbox if none of the answers is ok.

Section 2 – 30 points

Create a C# Form application similar with the one in the images below, by adding the appropriate controls and modifying the default properties values, that calculates and displays the result of length conversions from Metric to US values and display theirs results. The form should look like:

Add following controls to your form, and change/add the values to the properties:

Default name	Property	Setting
Form1	Name	frm_ConvertLegth
	Text	FirstName LastName – Student ID
label1	Text	Enter the value of the length you want to convert
label2	Text	Metric mm
label3	Text	US in
label4	Text	Metric cm
label5	Text	US in
label6	Text	Metric m
label7	Text	US yd
label8	Text	Metric km
label9	Text	US mile
label10	Text	Question 1st answer(s):
label11	Text	Question 2nd answer(s):
label12	Text	Question 3rd answer(s):
label13	Text	Question 4th answer(s):
label14	Text	Question 5th answer(s):
label15	Text	Student ID::
label16	Text	Total value of ID numbers:

textBox1 to 4	Name TextAlign TabIndex Text	txtBox1 Right 0, 0	txtBox2 2, 0	txtBox3 4, 0	txtBox4 6 0
textBox5 to 8	Name ReadOnly Text TextAlign	txtBox5 True 0	txtBox6 0	txtBox7 0	txtBox8 0
button1 to 4	Name Text TabIndex	btn1 &1st conversion ...	btn2 ...	btn3 ...	btn4 &4th conversion
textBox9 to 13	Name ReadOnly TabStop Text TextAlign	txtAns1 True False a, b, c, d Center	txtAns2 a, b, c, d	txtAns3 a, b, c, d	txtAns4 a, b, c, d
Button5	Name Text TabIndex	btnShow &Show Answers 8			
textBox14 to 20 textBox21	Name Name ReadOnly TabStop TextAlign	txtID1 txtTotalID True False Center	txtID2	txtID3	txtID4 txtID5 txtID6 txtID7
Button6	Name Text TabIndex	btnTotalID Show &Total ID 9			
button7	Name Text TabIndex	btnReset &Reset 10			
button8	Name Text TabIndex	btnExit &Exit 11			


Test your code and format the output similar with the one from the image.

Metric mm	<input type="text" value="1"/>	1st Conversion	<input type="text" value="0.03937"/>	US in
Metric cm	<input type="text" value="1"/>	2nd Conversion	<input type="text" value="0.3937"/>	US in
Metric m	<input type="text" value="1"/>	3rd Conversion	<input type="text" value="1.0936"/>	US yd
Metric km	<input type="text" value="1"/>	4th Conversion	<input type="text" value="0.6214"/>	US mile

Section 3 – 30 points

Add code to your C# Form application (business or middle level) to calculate and present the results following those rules:

Metric			US or Imperial
1 millimetre [mm]		➔	0.03937 in
1 centimetre [cm]	10 mm	➔	0.3937 in
1 metre [m]	100 cm	➔	1.0936 yd
1 kilometre [km]	1000 m	➔	0.6214 mile

Method	Description
frmAreaConvert_Load()	Sets the all textBox fields to their default values and default read only properties.
btn1_Click()	Performs the conversion from the entered value in Metric textBoxes to the US textBoxes. Change the property readOnly of Metric to true, and change the focus to the next Metric field. Use object(s) of ConvertLegth class.
btn2_Click()	Performs the conversion from the entered value in Metric textBoxes to the US textBoxes. Change the property readOnly of Metric to true, and change the focus to the next Metric field. Use object(s) of ConvertLegth class.
btn3_Click()	Performs the conversion from the entered value in Metric textBoxes to the US textBoxes. Change the property readOnly of Metric to true, and change the focus to the next Metric field. Use object(s) of ConvertLegth class.
btn4_Click()	Performs the conversion from the entered value in Metric textBoxes to the US textBoxes. Change the property readOnly of Metric to true. Use object(s) of ConvertLegth class.
btnTotalID_Click()	Set the result values for the sum of 7 numbers of your ID into read only txtTotalID.
	
btnShow_Click()	Sets the results values for the 5 questions to appropriate read only textBoxes.
btnReset_Click()	Sets the textBox fields to their default values and default read only properties.
btnExit_Click()	Close the application if the user click OK , and set the values and properties to their default values if the user click Cancel , use the interrogation logo into the MessageBox.

Use **try** and **catch** for the entered (converted) values.

Identify yourself, enter the current date and have a short description for the application as a comment on the top section of your work file.

Test your application, theirs functionalities, save the c# solution, compress the folder and send it by LEA of Omnivox.

Section 4 – 30 points

Create a C# console application where you will use the same class declaration (fields, methods, and properties); instantiate the class (create objects).

Store (save) all the entered values into appropriate data types **private** fields of the object(s) by using their **public** properties (**set**, **get**) or constructors of a **ConvertLegth** class, and then, present all the calculated information using **public** methods, in the proper format on console using placeholders.

Use **set** and **get** for the 4 entries and/or the appropriate methods and/or constructors for the presentation of the results (use placeholders).

Create a menu (use **do - while**) to present 4 different conversions options and the option to quit. Use **try** and **catch** for each conversion,

Use **Alt + PrintScreen** to save the testing results as an image and send it with the c# solution or add the output result as a comment.

Identify yourself, enter the current date and have a short description of your work as a comment on the top section of your C# file.

Test your application, theirs functionalities, save the C# solution, compress the folder and send it by LEA of Omnivox.

Thank you.