

Grade 12 Computer and Information Science  
Mid-Unit Quest – Introduction to C# and Review of Programming

Mr. N. Nolfi

Victim: Mr. Solutions*Well done Mr. S.!*

KU

APP

24/24

15/15

1. Match each term in the left column with the *best* definition in the right column. (16 KU)

M ✓ index~~A~~ Operators, such as &&, || and ! that are used to create compound conditions.Q ✓ braces~~B~~ An appliance that corrects dental irregularities.K ✓ assignment statement~~C~~ Something students hate to get from their teachers.F ✓ primitive data type~~D~~ A method of conveying information used by “cave people.”I ✓ %~~E~~ A function or “action” that belongs to an object.P ✓ !~~X~~ A data type that is not defined in terms of simpler types.U ✓ object~~G~~ A tangible and visible entity.L ✓ repetition (looping)~~H~~ Any time during which a program is being executed.J ✓ data field (property)~~J~~ Operator used to evaluate the remainder obtained upon dividing two integers.S ✓ declaration~~J~~ A variable that belongs to an object.A ✓ conditional operators~~K~~ A statement that is used to give a value to a variable.E ✓ method~~J~~ A programming structure that allows a particular group of statements to be repeated a certain number of times or while a certain condition is true.R ✓ selection (“if”)~~M~~ A number that is used to identify a particular character of a string.H ✓ run-time~~N~~ The time at which the police show up and catch you in the act!T ✓ compile~~S~~ Any time during which a program’s source code is being edited.O ✓ design-time~~P~~ The “not” operator used in C-style languages.~~Q~~ Symbols used to enclose a group of statements that are to be treated as a single statement.~~R~~ A programming structure that allows a particular group of statements to be executed while other groups of statements may be ignored.~~S~~ A statement that specifies the name, data type and other aspects of a variable.~~T~~ Translate a high-level program to bytecode, assembly code or machine code.U Generally a collection of properties (data fields) and methods. In the .NET environment, events are also included.

2. Translate into C# assignment statements. (8 KU)

(a) Calculate the number of whole <i>hours</i> in a given number of seconds.	$hours = seconds / 3600;$ (works as long as "hours" and "seconds" are integers)
(b) $A = \pi r^2 + \pi r s$	$coneArea = Math.PI * Math.Pow(radius, 2) + Math.PI * radius * slantHeight;$

3. Consider the following C# code: (2 APP)

```
long b = 9;
```

```
int a = b; X
```

Is this valid C# code? If so, explain why. If not, explain why it isn't and make corrections.

Since implicit conversions are not allowed in C#, a "long" value (64-bit integer) cannot be assigned to an "int" variable (32-bit integer).

Correction

```
int a = (int)b;
```

OR long a = b; OR

```
int b=9;
int a=3;
```

4. Explain how the following C# code could be improved: (3 APP)

```
if (ferrariRadioButton.Checked)
    make = "Ferrari";
else if (lamborghiniRadioButton.Checked)
    make = "Lamborghini";
else if (bugattiRadioButton.Checked)
    make = "Bugatti";
else if (alfaRomeoRadioButton.Checked)
    make = "Alfa Romeo";
```

This should be written as 1 "if" statement with 4 clauses

since only ONE of the radio buttons can be selected.

Correction if ( ..... )  
else if ( ..... )  
etc.

5. For the given code snippet, create a memory map and state the problem that is solved. (10 APP)

Code Snippet	Memory Map	Problem that is Solved																										
<div>Values before Entering Loop</div> <pre>// Recall that Math.Pow(2, i) means // "2 to the exponent i." The cast // operator (int) is needed to force // a conversion from "double" to "int"  int sudman=1;  for (int i=1; i&lt;=10; i++) {     if (i % 2 == 0)         sudman *= (int)Math.Pow(2,i); }</pre> <div>Values after Exiting Loop</div>	<table><tr><th>i</th><th>sudman</th></tr><tr><td>-</td><td>1</td></tr><tr><td>1</td><td>1</td></tr><tr><td>2</td><td>4</td></tr><tr><td>3</td><td>4</td></tr><tr><td>4</td><td>64</td></tr><tr><td>5</td><td>64</td></tr><tr><td>6</td><td>4096</td></tr><tr><td>7</td><td>4096</td></tr><tr><td>8</td><td>1048576</td></tr><tr><td>9</td><td>1048576</td></tr><tr><td>10</td><td>1073741824</td></tr><tr><td>-</td><td>1073741824</td></tr></table>	i	sudman	-	1	1	1	2	4	3	4	4	64	5	64	6	4096	7	4096	8	1048576	9	1048576	10	1073741824	-	1073741824	<p>By the time the loop has finished executing, the variable "sudman" stores...</p> <p><u>the value obtained when the even powers of 2 between 1 and 10 are multiplied (i.e. <math>2^2 \times 2^4 \times 2^6 \times 2^8 \times 2^{10}</math>)</u></p>
i	sudman																											
-	1																											
1	1																											
2	4																											
3	4																											
4	64																											
5	64																											
6	4096																											
7	4096																											
8	1048576																											
9	1048576																											
10	1073741824																											
-	1073741824																											

Happens to be  $2^{30}$