

Exercises

<https://csci-1301.github.io/about#authors>

September 5, 2023 (05:12:31 PM)

Contents

1 Homework #1	1
1.1 Part I — Questions	1
1.2 Part II – Problems	2
2 Homework #2	3
2.1 Part I — Questions	3
2.2 Part II – Problems	3
3 Homework #3	6
3.1 Part I – Questions	6
3.2 Part II – Problems	8
4 Homework #4	9
4.1 Part I — Questions	9
4.2 Part II – Problems	11
5 Homework #5	12
5.1 Part I — Questions	12
5.2 Part II – Problems	13
6 Homework #6	13
6.1 Part I — Questions	13
7 Quizzes	15
7.1 Quiz 1	15
7.2 Quiz 2	15
7.3 Quiz 3	16

1 Homework #1

1.1 Part I — Questions

1. List five pieces of software, and three hardware components of a computer.
2. What is a GUI?
3. What, if any, is the difference between a compiler and an assembler?
4. Give a specific characteristic of C# compared to other programming languages.

2 Homework #2

2.1 Part I — Questions

1. In C#, what is the “escape character”? Why is it useful?
2. Suppose you replace every `*` in your program with the `!` symbol. Are you completely sure that your program would still compile? Why or why not?
3. Give the values of `a` and `b` after the following four instructions have been executed.

```
int a, b;  
a = 4;  
b = a * 3 + 1;  
a /= 2;
```

4. Is there an error in the following code? Explain the error or give the value of `b` after the second statement is executed.

```
float a = 3.7f;  
int b = (int)a;
```

5. There is an error in the following code, at the second line. Explain the error, and how you could fix this second line using a cast operator, without changing the datatype of the `b` variable.

```
decimal a = 2.5M;  
int b = a / 2;
```

6. What is the return type of the operation `12.4 * 3`?
7. Write a statement that performs an implicit conversion between two different numeric datatypes.
8. In C#, what is the name of the method used to read input from the user?
9. Will those statements, if placed in a proper `Main` method, compile? Could this program crash at execution time? Justify your answer.

```
int myAge;  
Console.WriteLine("Please enter your age:");  
myAge = int.Parse(Console.ReadLine());
```

10. Write a series of statements that: a) Declare an `string` variable named `favoriteColor`; b) Display on the screen a message asking the user to enter his or her favorite color; c) Read the value entered by the user and store it in the `favoriteColor` variable. You can combine some of the statement(s) if you want, but do not display at the screen any information that was not explicitly asked.

2.2 Part II – Problems

The following three exercises **do not** require a computer. Make sure you feel ready before starting them, try to do them with limited time and without notes, and, if you want, check your answer using your IDE.

1. This problem restates the content of the Order of Operations¹ section of the lecture notes and ask you to answer various problems.

There are 5 different arithmetic operations available in C#:

¹<https://csci-1301.github.io/book.html#order-of-operations>

Operation	Arithmetic Operator	Algebraic Expression	Expression
Addition	+	$x + 7$	<code>myVar + 7</code>
Subtraction	-	$x - 7$	<code>myVar - 7</code>
Multiplication	*	$x \times 7$	<code>myVar * 7</code>
Division	/	$x/7$, or $x \div 7$	<code>myVar / 7</code>
Remainder (a.k.a. modulo)	%	$x \bmod 7$	<code>myVar % 7</code>

Computing operations involving one of them is straightforward:

Operation	Result
<code>3 + 4</code>	<code>7</code>
<code>3 - 4</code>	<code>-1</code>
<code>3 * 4</code>	<code>12</code>
<code>6 / 2</code>	<code>3</code>
<code>6 % 4</code>	<code>2</code>

But things can get complicated when multiple operators are used, but no parenthesis are indicated. For instance, should

`7 / 2 - 4 * 8 % 3`

be read as

$$\begin{aligned}
 (7 \div 2) - ((4 \times 8) \bmod 3) &= 3.5 - (32 \bmod 3) \\
 &= 3.5 - 2 \\
 &= 1.5
 \end{aligned}$$

or as

$$\begin{aligned}
 (7 \div (2 - 4)) \times (8 \bmod 3) &= (7 \div (-2)) \times 2 \\
 &= (-3.5) \times 2 \\
 &= -7
 \end{aligned}$$

Certainly, the result is not the same and there are other possible ways this calculation may be performed!

Actually, C# uses the following three rules:

- a) *, /, and %, called the “multiplicative operations,” are always evaluated before + and -, called the “additive operations.” So that, for instance,

`2 - 4 * 8`

will be evaluated as $2 - (4 * 8) = -30$.

- b) If there are multiple operations of the same type, they are evaluated from left to right. For instance,

`4 / 2 * 8`

will be evaluated as $(4 \div 2) \times 8 = 16$ and

`4 - 2 + 8`

will be evaluated as $(4 - 2) + 8 = 10$.

- c) Parenthesis can be used to force a particular order of evaluation, so that $2 * (3 + 4)$ will be evaluated as $2 \times (3 + 4) = 2 \times 7 = 14$, not as $(2 * 3) + 4 = 6 + 4 = 10$ as it would without the parenthesis.

Answer the following:

- a) Which of the following operation(s) compute the arithmetic expression $(x \times (3 \bmod 5)) - (y \times 7)$?

- i. $x * 3 \% 5 - y * 7$
- ii. $x * (3 \% 5) - y * 7$
- iii. $(x * 3) \% 5 - y * 7$
- iv. $x * 3 \% (5 - y * 7)$
- v. $(x * 3 \% 5) - (y * 7)$
- vi. $(x * ((3 \% 5) - (y * 7)))$

- b) State the order of evaluation of the operators in each of the following operations, and compute the resulting value:

- i. $8 - 39 * 1 / 12 + 5$
- ii. $12 + -23 / 12 \% 3$
- iii. $90 * 23 / 34 - 12 - 13$
- iv. $12 \% 83 - 2 * 3$

- c) (Optional) Check your answers using your IDE. You can use a statement of the form:

```
Console.WriteLine($"8 - 39 * 1 / 12 + 5 is {8 - 39 * 1 / 12 + 5}");
```

2. Write down, on a piece of paper, a fully compilable program that initializes an **int** variable named **persons** with the value 5, an **int** variable named **bottles** with the value 3, and a **double** variable named **literPerBottle** with the value 1.5. What should be the type of the **literPerPerson** variable to be able to be assigned the number of liters every person is going to get, if split equitably? Write the correct initialization of that variable and a statement that displays its value.

Place a delimited comment with a your name and the time at which you wrote the program at the top of the program.

3. Write down, on a piece of paper, a program that:

- a) Declares a **string** variable named **userName**.
- b) Displays on the screen "Please enter your name, followed by enter:".
- c) Reads a **string** value from the keyboard and assigns the value to the **userName** variable.
- d) Declares an **int** variable named **number**.
- e) Displays on the screen "Please enter your number:".
- f) Reads an **int** value from the keyboard and assigns the value to the **number** variable.
- g) Declares a **string** variable named **id** and initializes it with the string referenced by the **userName** variable, followed by the number entered by the user (you can concatenate a string and an int using the **+** sign).
- h) Displays on the screen, "Your id is" and the content of the **id** variable.

Here is an example of execution, where the user input is underlined, and hitting "enter" is represented by \leftarrow :

```
Please enter your name, followed by enter.
```

```
Jaylah $\leftarrow$ 
```

```
Please enter your area code, followed by enter.
```

```
49391 $\leftarrow$ 
```

```
Your id is Jaylah49391
```

```
Press any key to continue . . .
```

3 Homework #3

3.1 Part I – Questions

1. What is “an instance of a class”?
2. Give two access modifiers.
3. Write a statement that creates a new object from the `Rectangle` class.
4. Do different objects from the same class share their instance variables?
5. Suppose we have a `Circle` class containing

```
public void SetRadius(double radiusArgument)
{
    radius = radiusArgument;
}
```

Write a statement that create a `Circle` object, and one statement that sets its radius to 3.5.

6. Indicate the order of evaluation of the operators in each of the following C# operations by adding parenthesis or developing the expression one step at a time, and compute the resulting value:
 - a) `3 * 4 - 2`
 - b) `3 % 2 + 3`
 - c) `2 - 3 + 3 * 2`
 - d) `2 + 2 * 1 - 4`
7. Write the complete implementation of a class that contains two attributes (with different data types), a setter for one attribute, a getter for the other attribute, a custom constructor, and a `ToString` method. You can re-use an example from a lecture or a lab, as long as it satisfies those conditions, or you can invent one. No need to write an application program.
8. What does the keyword `return` do?
9. Write a getter for an attribute of type `string` named `myName`.
10. Write a setter for an attribute of type `int` named `myAge`.
11. Assuming `name` is a `string` instance variable, there is problem with the following setter. Fix it.

```
public int SetName(string val){
    name = val;
}
```
12. Briefly describe what a format specifier is. Write a statement that uses one.
13. Write a method for the `Rectangle` class that divides the length and width of the calling object by a factor given as a parameter.
14. Draw the UML diagram of a class named “Student” with a single attribute, “name”, of type `string`, and two methods, `SetName` and `GetName`.
15. Write a `ToString` method for a `Account` class with two attributes, a `string` attribute called `name` and a `decimal` attribute called `amount`.

Circle

16. Consider the following UML diagram:

Circle

- radius : float
+ setRadius(radiusParam : float) : void
+ getRadius(): float
+ getArea(): float

What is the name of the class, what are the methods and attributes of the class?

17. Is it possible to have more than one constructor defined for a class? If yes, how can C# know which one is called?
18. Write a constructor for a **Soda** class with one **string** attribute called **name**.
19. Assume we have a **Polygon** class, that have only one attribute, an **int** called **numberOfSides**. Write a constructor for that class.
20. What is the “default” constructor? Do we always have the possibility of using it?
21. Consider the following partial class definition:

```
public class Book
{
    private string title;
    private string author;
    private string publisher;
    private int copiesSold;
}
```

- a) Write a statement that would create a **Book** object.
- b) Write a “getter” and a “setter” for the **title** attribute.
- c) Write a constructor for the **Book** class taking at least one argument (you are free to decide which one(s)).
22. Consider the following partial class definition:

```
class DVD
{
    private string title;
    private decimal price;
}
```

- a) Write a “setter” for the **title** attribute.
- b) Write a constructor for the **DVD** class that takes two arguments.
- c) Write a method called **Discount** that decreases the **price** attribute by 20.55%.
- d) Write a (good, informative) **ToString** method for the class.
- e) Write statements that ask the user to enter a price and then create a **DVD** object with a **price** attribute equal to the price the user entered. (The object’s **title** attribute can be anything you choose).
- f) Draw the UML class diagram for the class you obtained by adding the above four methods to our original class definition.
23. Consider the following partial class definition:

```

class Book
{
    private string title;
    private decimal price;
}

```

- a) Write a “getter” for the `title` attribute.
- b) Write a constructor for the `Book` class that takes two arguments.
- c) Write a method called `AddTaxes` that increases the `price` attribute by 6.35%.
- d) Write a (good, informative) `ToString` method for that class.
- e) Write statements that ask the user to enter a price and then create a `Book` object with a `price` attribute equal to the price the user entered. (The object’s `title` attribute can be anything you choose).
- f) Draw the UML class diagram for the class you obtained by adding the above four methods to our original class definition.

24. Assume that my `Pet` class contains one custom constructor:

```

public Pet(string nameP, char genderP){
    name = nameP;
    gender = genderP;
}

```

What is the problem with the following statement?

```

Pet myPet = new Pet('M', "Bob");

```

3.2 Part II – Problems

There is only one problem this time, and it is harder than what you’ll be asked to do during the exam. Being able to solve it is an excellent sign that you are ready.

1. You are going to design a class named `Triangle`. A triangle has three angles, but knowing the value of only two angles is sufficient to determine the value of the third, since they always add up to 180°. Hence, it is sufficient to have only two `double` attributes, `angle1` and `angle2`. We want to define several methods:
 - a no-arg constructor that sets the value of `angle1` to 60.0 and the value of `angle2` to 60.0,
 - another constructor, that takes two arguments, and assigns to `angle1` the value of the first argument, and assigns to `angle2` the value of the second argument,
 - getters for `angle1` and `angle2`,
 - a method that computes and returns the value of the third angle, that we name `ComputeAngle3`,
 - a method that rotate the triangle: the value of the first angle should be replaced with the value of the second angle, and the value of the second angle should be replaced with the value of the third angle.
- a) Write the UML diagram for the `Triangle` class.
- b) Write the full, compilable implementation of the `Triangle` class.

4 Homework #4

4.1 Part I — Questions

1. What is sequential processing?

2. Decide if the following boolean expressions will evaluate to **true** or **false**:

- a) `3 > 2.0 && false`
- b) `(4 != 3) || false`
- c) `'A' == 'b' && ! false`
- d) `(! false) == (true || 4 == 3)`

3. Decide if the following Boolean expressions will evaluate to **true** or **false**:

- a) `3 > 2.0 || true`
- b) `(4 != 3) && false`
- c) `'A' == 'b' || ! false`
- d) `(! true) == (true || 4 != 3)`

4. For each of the following Boolean expressions, decide if it will evaluate to **true** or **false**:

- a) `('y' == 'Y') && true`
- b) `6 + 2 < 8 || 3 > 4`
- c) `(true && 4 == 3) == false`
- d) `4 > 4 && !false`

5. For each of the following Boolean expressions, decide if it will evaluate to **true** or **false**:

- a) `('y' != 'Y') && true`
- b) `6 + 2 < 12 || 3 > 4`
- c) `(true && 4 >= 3) == false`
- d) `13 <= 4 * 3 || !false`

6. What relational operator is used to determine whenever two values are different?

7. Give three relational operators, and then two logical operators.

8. Is there a simpler way to write the expression `over21 == true`, assuming that `over21` is a Boolean variable?

9. In C#, is there a difference between `=` and `==`? Write a statement that uses `=`.

10. Is the following statement correct, i.e., would it compile, assuming `myFlag` is a **bool** variable, and `myAge` is an initialized **int** variable?

```
if ( myAge > 20 )
{
    myFlag = true
};
```

11. Write an **if** statement that prints “Bonjour !” if the value of the **char** variable `lang` is `'f'`.

12. For each of the following boolean expressions, decide if it will evaluate to **true** or **false** when the boolean variables `x`, `y` and `z` are all set to **true**:

- `x || y && z`
- `!x || y && z`
- `!(x || y) && (z && y)`
- `(!x && x) || (!x || x)`

Do the same when they are all set to `false`.

13. Write a `boolean` expression that evaluates to `true` if a variable `x` is between 3 (excluded) and 5 (included).
14. Write an `if-else` statement that assigns `"Minor"` to an already declared `string` variable `legalStatus` if `age` is strictly less than 18, and that assigns `"Major"` to `legalStatus` otherwise.
15. Write an `if-else` statement that displays "It's free for you!" if an `int` variable `age` is between 0 and 18, and "It's \$5.00." otherwise.
16. Assume we initialized an `int` variable called `courseNumber` and a `string` variable called `courseCode`. Write a series of statements that will display:
 - a) "I'm taking this class!" if `courseNumber` is 1301 and `courseCode` is CSCI;
 - b) "That's my major!" if `courseCode` is CSCI;
 - c) "Is that an elective?" if `courseNumber` is greater than 3000; or
 - d) "Is it a good class?" otherwise.

Your program should display exactly one message.

17. Assume we previously initialized an `int` variable called `graduationYear` and a `string` variable called `graduationSemester`. Write a series of statements that will display:
 - a) "I will graduate at the same time!" if `graduationYear` is 2023 and `graduationSemester` is Fall;
 - b) "I love this season." if `graduationSemester` is Spring;
 - c) "That is in a long time!" if `graduationYear` is greater than 2025; or
 - d) "I hope you'll have an in-person ceremony!" otherwise.

Your program should display exactly one message.

18. Assume we previously initialized a `char` variable called `myChar`. Write a series of statements that will display if the character is...
 - a) Uppercase
 - b) Lowercase
 - c) A number
 - d) or none of those.

Your program should display exactly one message. Bonus: Make your message also display the ASCII value of the character.

19. What will be displayed on the screen by the following program?

```
int x = 3, y = 2, z = 4;
if (x > y) {z += y;}
if (x > z) {y -= 4;}
Console.WriteLine($"x is {x}, y is {y}, and z is {z}.");
```

20. (*We'll use the 24-hour clock, sometimes called "military time".*) Assuming that an `int` variable `hour` has been initialized, write part of a program that would display on the screen "Good morning" if `hours` is less than or equal to 12, and "Hello" otherwise.
21. Assuming that `myString` is a string variable, write a statement that prints "Hello, Melody!" if the value of `myString` is equal to `Melody`, and nothing otherwise.
22. What will be displayed on the screen by the following program?

```
int x = 3, y = 2, z = 4;
if (y >= z) {z += y;}
else if (x != y) { if (false) {z -= 3;} else {z += x;}}
Console.WriteLine($"x is {x}, y is {y}, and z is {z}.");
```

23. Give an example of an `if` statement that could not be rewritten as a `switch`.

4.2 Part II – Problems

This time, the two exercises **do not** require a computer, and are here to craft on your problem-solving skills. Make sure you feel ready before starting them, try to do them with a limited amount of time and without notes, and check your answer using your IDE.

1. Write a program that asks the user to write a country name and stores the user's input into a string variable. Then, compare that string with `"france"`: if it is equal, then display at the screen `"Bienvenue en France !"`. Then, compare that string with `"usa"`: if it is equal, then display at the screen `"Welcome to the US!"`. If the string is different from both `"france"` and `"usa"`, then display at the screen `"Welcome to"` followed by the name of the country the user typed in. Can you think of two ways to implement this program, one using `if-else-if` statements, the other using `switch`?
2. You want to write a small program for an on-line printing company. Your program should ask the user to chose a format (10×15 centimeters, or 8×11 inches), ask if it is the first time the customer order through your company, and a number of copies. Then, calculate the total cost of printing those pictures, knowing that
 - Printing a 10×15 centimeters picture costs \$0.20, printing a 8×11 inches picture costs \$0.25,
 - A new customer gets a \$3 coupon if the order is more than \$5,
 - A 10% discount is given if more than 50 copies were ordered,
 - The two previous offers can be cumulated.

Display on the screen a message starting by "Welcome!", then a new line, then "We cherish our new customers" if it is the first time the user uses your company, " , so we're giving you a \$3 discount!" if the user is allowed to get the coupon, then print the total and "You had a 10% discount!" if the user ordered more than 50 copies. See below for examples of execution, where the user input is underlined, and hitting carriage return is represented by `↵`.

```
Enter 'c' for 10x15cm, anything else for 8x11in.
c ↵
Is this your first time here? Type 'y' for 'yes'.
y ↵
Enter a number of copies.
90 ↵
Welcome!
We cherish our new customers, so we are giving you a $3 discount!
Your total is $13.50. You had a 10% discount!

Enter 'c' for 10x15cm, anything else for 8x11in.
p ↵
Is this your first time here? Type 'y' for 'yes'.
Not at all ↵
Enter a number of copies.
120 ↵
Your total is $27.00. You had a 10% discount!
```

5 Homework #5

5.1 Part I — Questions

1. Assume you are given an un-assigned `string` variable `letterGrade`, and an already assigned `float` variable `numberGrade`. Write a small program that assigns "A" to `letterGrade` if `numberGrade` is between 100 and 90 (both included), "B" to `letterGrade` if `numberGrade` is between 90 (excluded) and 80 (included), etc., and "Invalid data" if `numberGrade` is strictly lower than 0 or strictly greater than 100. Should you use a `switch` statement or a `if ...else if ...else`?
2. Given an `int` variable `counter`, write three statements to decrement its value by 1.
3. What will be displayed on the screen?

```
int x = 3, y = 7;
Console.WriteLine (x++ +" and "+ --y);
```

4. What will be displayed on the screen by the following program? Write the spaces and new line explicitly.

```
int counter = 10;
while (counter > 5)
{
    counter--;
    Console.Write(counter + "\n");
    if (counter == 7) {
        Console.WriteLine("Bingo");
    }
}
```

5. What will be displayed on the screen by the following program?

```
int counter = 10;
while (counter != 5) ;
Console.Write(counter + "\n");
counter--;
```

6. What is input validation? Name a control structure that can be used to perform it. Why is it important?
7. What is a sentinel value?
8. Write a small program that asks the user for an integer, and displays "It is positive" if the number entered is positive, "It is negative" if the number entered is negative, and "Not a number" if the user entered a string that is not an integer.
9. Write a program containing a `while` loop that would display the numbers between -100 and 100 (both included) with a space between them when executed.
10. Assume you are given an initialized `string` variable `name`, and a `string` variable `field`. Write a small program that assigns to `field`
 - "CS" if `name` is "Turing" or "Liskov",
 - "Math." if `name` is "Aryabhata" or "Noether",
 - "Unknown" otherwise.
11. Write a program that asks the user to enter a value between 1900 and 1999 (both included), and asks again as long as the user enters integers outside that range.

5.2 Part II – Problems

1. Write a `switch` statement that calculates the number of days in a particular month. You should assume that you are given already assigned `month` and `year` `int` variables, and that your program should set an already declared `int` `numberOfDays` variable to 28, 29, 30 or 31 depending on the month / year combination. Your program should start with a `switch` matching `month` against certain values, and, if `month` is 2, uses an `if` statement to decide whenever the number of days is 28 or 29. You can use something like

```
switch (month) {  
    ...  
    case (2):  
        if ...  
        ...  
        break;  
    ...  
}
```

6 Homework #6

6.1 Part I — Questions

1. Write a statement that creates a 10-element `int` array named `numbers`.
2. In the following, what is the value of the size declarator? What is the value of the index?

```
int[] numbers;  
numbers = new int[8];  
numbers[4] = 9;
```

3. Draw the content of the `scores` array once those statements have been executed.

```
int[] scores = new int[3];  
scores[0] = 13;  
scores[2] = 25;
```

4. What will be displayed on the screen by the following program?

```
for (int num = 3 ; num <= 5 ; num++)  
    Console.Write(num + " ");
```

5. Write a 'for' loop that displays on the screen the sequence "1, 2, 3, 4, 5, 6, 7, 8, 9, 10, ".
6. Write a 'for' loop that displays on the screen the sequence "1, 2, 3, 4, 5, 6, 7, 8, 9, 10", (note that there is no comma after 10).
7. Write a 'for' loop that displays on the screen the sequence "1 3 5 7 9 ".
8. Given an `int` variable `myVar` initialized with a positive value, write a loop that sums the integers between 0 and `myVar` (i.e., $0 + 1 + \dots + (\text{myVar} - 1) + \text{myVar}$).
9. Consider the following code:

```

for (int y = 1; y <= 3; y++)
{
    for (int z = 1; z < 5; z++)
        Console.WriteLine("Scene " + y + ", take " + z + ". " );
    Console.WriteLine();
}

```

How many times does the outer loop iterate (i.e., how many scenes are shot)? How many times does the inner loop iterate (i.e., how many takes for each scene)? Finally, what is the total number of iterations of the nested loops (i.e., how many takes are made, total)?

10. What will be displayed on the screen by the following code?

```

int[] values = new int[6];
for (int i = 0 ; i < 6 ; i++)
    values[i] = (i*2);
foreach (int j in values)
    Console.WriteLine(j);

```

11. Suppose we are given an `int` array `dailyPushUp` with 7 elements. Write a piece of code that display the value of the elements stored in the array `dailyPushUp`.
12. What is “array bounds checking”? When does it happen?
13. What would be the size of the `test` array after the following statement has been executed?

```
int[] test = {3, 5, 7, 0, 9};
```

14. Write a statement that creates and initializes a `double` array with the values 12.5, 89.0 and 3.24.
15. What is the value of `count` and the content of `number` once the following has been executed?

```

int count=2;
int[] number={3, 5, 7};
number[count--] = 8;
number[count]--;

```

16. Describe what the following code would do.

```

int[] record = { 3, 8, 11 };
int accumulator = 0;
foreach (int i in record)
    accumulator += i;

```

17. Assuming we are given an `int` array named `arrayF`, write a program that adds one to each of its elements. That is, if `arrayF` contains 3, 5, 7 and -2 before your program is executed, it should then contain 4, 6, 8 and -1 after your program was executed.
18. Assuming we are given an `int` array named `arrayF`, write a program that displays the product of its elements. That is, if `arrayF` contains 2, 3 and -1, then your program should display -6.
19. Write a static method (header included) that takes as argument an `int` array, and display on the screen the value of each element of that array.
20. Write a static method (header included) that takes as argument an `int` array, and stores the value 10 in each element of that array.

7 Quizzes

Those quizzes are given as *examples*, to help you practise. They were given at week 4 and 7.

7.1 Quiz 1

1. (3 pts) List three keywords.
2. (4 pts) For each of the following, indicate if they are a “rule” of C# or a “convention” between programmers by ticking the appropriate column. The first answer is given as an example.

Statement	Rule	Convention
Code should be commented.		✓
Case matters.		
Variable names should be descriptive.		
Keywords cannot be used as identifiers.		
Each “.cs” file should contain exactly one class.		

3. (4 pts) Write a statement that would display, “Hi Mom!” (*with the quotes*) followed by a new line on the screen.
4. (Bonus) Give examples of situations where the adage “*Spaces and new lines do not matter in programs*” is actually erroneous.

7.2 Quiz 2

1. (2 pts) What is the relational operator used to determine whenever two values are equal?
2. (5pts) Write a **boolean** expression that evaluates to **true** if a variable **x** is between -10 (excluded) and 10 (included).
3. (3 pts) What will be displayed on the screen by the following program?

```
int x = 5, y = 1, z = 2;
if (x != z || y < z ) {x += y;}
if (z > y) {if (z % 2 == 0) {z -= 3;} else {z += x;}} else {x = y + z;}
Console.WriteLine($"x is {x}, y is {y}, and z is {z}.");
```

4. (8 pts) Assume we initialized a **string** variable named **month** and a **double** variable named **temperature**. Write a series of statements that will display *exactly* one of the following messages: “What a nice summer day!” if **month** is “July” and **temperature** is less than 90 (included); “Better wear a jacket” if **temperature** is between 45 and 60 (both included); “Happy holidays!” if **month** is “December”; or “Have a nice day” otherwise.
5. (Bonus) Give a program that displays “Leap year” if a **year** variable is
 - a) divisible by 4; and
 - b) not divisible by 100, unless it is also divisible by 400.

Your program should correctly identify 2000 and 2400 as leap years, and 1800, 1900, 2100, 2200, 2300, or 2500 as *not* leap years.

7.3 Quiz 3

1. (7 pts) Write a `switch` statement that sets a `decimal price` variable to 1.50 if a string variable `iceCreamFlavor` contains `"vanilla"`, 1.75 if it contains `"chocolate"` or `"mint"`, and -1 otherwise.
2. (2 pts) Write a statement that applies the increment operator in prefix position to a variable `test`.
3. (8 pts) Write a `while` loop that displays "1 2 3 4 5 6 7 8 9 10 11 12 13" (spaces included!) at the screen.
4. (+3+2 pt, bonus) Give a program that displays every multiple of 3 between 0 and 10,000 (that is, "0 3 6 9 12 ... 9999"). Bonus (again!): display how many such numbers there are in total.