



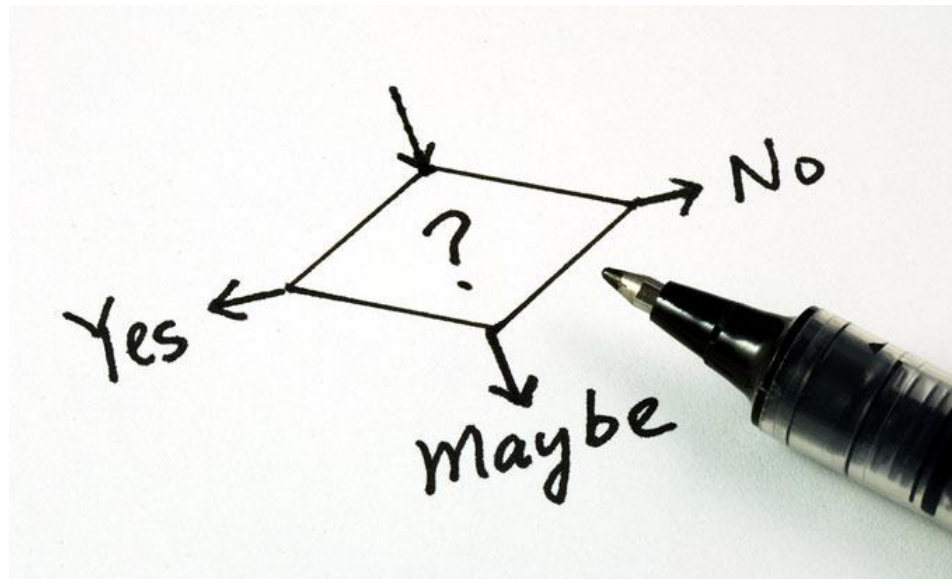
# LAB 3: DECISION MAKING

BME 121, Fall 2016

Rasoul Nasiri

# Topics

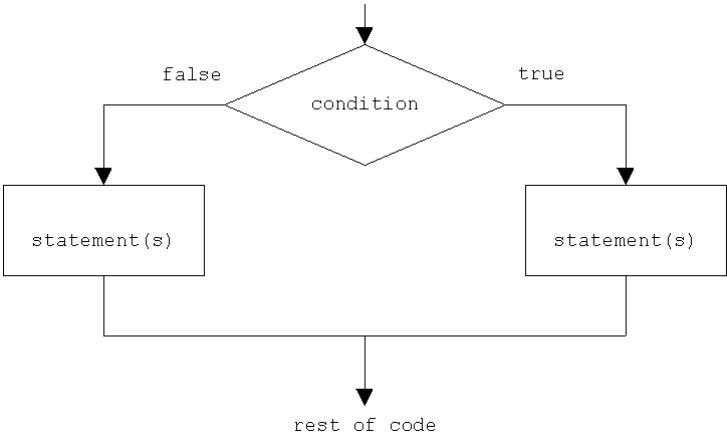
- Logical variable and comparisons
- More example with If-else
- Logical operators
- Switch-case statement
- Random generator
- Programming Assignment 2



# Review of Boolean variable

## Relational Operators

Operators	Meaning	Example	Result
<	Less than	5<2	False
>	Greater than	5>2	True
<=	Less than or equal to	5<=2	False
>=	Greater than or equal to	5>=2	True
==	Equal to	5==2	False
!=	Not equal to	5!=2	True

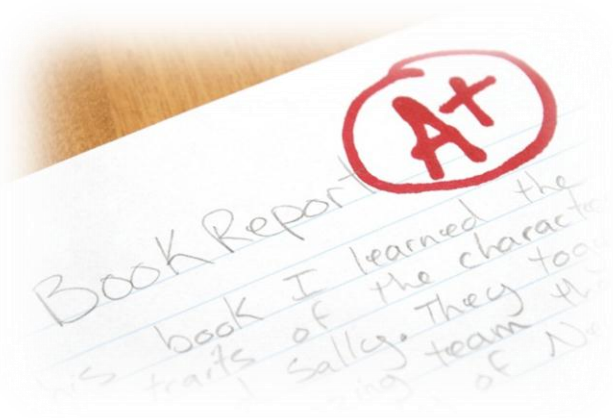


<b>&amp;&amp;</b>	<b>And</b>
<b>  </b>	<b>Or</b>
<b>^</b>	<b>Exclusive or</b>
<b>!</b>	<b>Not</b>

a	b	a && b	a    b	a ^ b	!a
false	false	false	false	false	true
false	true	false	true	true	true
true	False	false	true	true	false
true	true	true	true	false	false

# Practice:

- Write a program to map a score to grade letter
  - Score is an integer between 0 to 100
  - Score intervals corresponding to grade letters:
    - A: [80-100],
    - B: [70,80),
    - C: [60, 70),
    - D: [50, 60),
    - F: [0,50).



- Code: `gradeLetter.cs`

# If, if-else, if-else-if: flow of the program execution

- IF-else

95	
1	<code>double palScore = 95;</code>
2	<code>string gradeLetter; // Gradeletter correspond to palScore</code>
3	<code>if(score &gt;= 80)</code>
4	<code>{</code>
5	<code>    gradeLetter = "A";</code>
6	<code>    // ... any number of statements can be here</code>
	<code>}else</code>
	<code>{</code>
	<code>    gradeLetter = "Not A";</code>
	<code>}</code>
7	<code>WriteLine("Your gradeletter is {0}",gradeLetter);</code>

# If, if-else, if-else-if: flow of the program execution

- IF-else

75	95
1	1
2	2
3	3
	4
	5
4	6
5	
7	7

```
double palScore = 95;  
string gradeLetter; // Gradeletter correspond to palScore  
  
if(score >= 80)  
{  
    gradeLetter = "A";  
    // ... any number of statements can be here  
}  
else  
{  
    gradeLetter = "Not A";  
}  
WriteLine("Your gradeletter is {0}",gradeLetter);
```

# If, if/else, if-else-if: flow of the program execution

- IF-IF-IF

```
95 public void Main( )  
1  double palScore = 95;  
2  string gradeLetter; // Gradeletter correspond to palScore  
3  if(score >= 80) ←  
4  {  
5      gradeLetter = "A";  
6      // ... any number of statements can be here  
7  }  
8  if(palScore >= 70 && score < 80) ←  
9  {  
10     gradeLetter = "B";  
11 }  
12 if(palScore >= 60 && palScore < 70) ←  
13 {  
14     gradeLetter = "C";  
15 }  
16 WriteLine("Your gradeletter is {0}", gradeLetter);  
17 }
```

All the conditions  
would be checked

# If, if/else, if-else-if: flow of the program execution

- IF-IF-IF: Program should check all the conditions

75	95
1	1
2	2
3	3
	4
	5
4	6
5	
6	7
7	8

```
public void Main( )  
  
    double palScore = 95;  
    string gradeLetter; // Gradeletter correspond to palScore  
  
    if(score >= 80)  
    {  
        gradeLetter = "A";  
        // ... any number of statements can be here  
    }  
  
    if(palScore >= 70 && score < 80)  
    {  
        gradeLetter = "B";  
    }  
  
    if(palScore >= 60 && palScore <70)  
    {  
        gradeLetter = "C";  
    }  
  
    WriteLine("Your gradeletter is {0}",gradeLetter);  
}
```



# If, if/else, if-else-if: flow of the program execution

- IF-Else-IF: Program should check all the conditions

75	95	
1	1	<code>ic void Main( )</code>
2	2	<code>double palScore = 95;</code>
		<code>string gradeLetter; // Gradeletter correspond to palScore</code>
3	3	<code>if(score &gt;= 80)</code>
	4	<code>{</code>
	5	<code>    gradeLetter = "A";</code>
		<code>    // ... any number of statements can be here</code>
4		<code>} else if(palScore &gt;= 70 &amp;&amp; score &lt; 80)</code>
5		<code>{</code>
		<code>    gradeLetter = "B";</code>
		<code>}</code>
		<code>else if(palScore &gt;= 60 &amp;&amp; palScore &lt; 70)</code>
		<code>{</code>
		<code>    gradeLetter = "C";</code>
		<code>}</code>
6	6	<code>WriteLine("Your gradeletter is {0}", gradeLetter);</code>
		<code>}</code>

## Practice: Second Min

- Write a C# program that finds the second minimum in a series of 5 numbers.
  - Ask the user to enter numbers one by one and at the end you have to display second smallest.
  - You don't need to keep all the numbers, just keep 2 smallest ones.
  - For example the second smallest value in the collection of {4, 12, 5, 65, 1} is 4 when the min is 1. So your program should return 4
- 
- Code: `secondMin.cs`

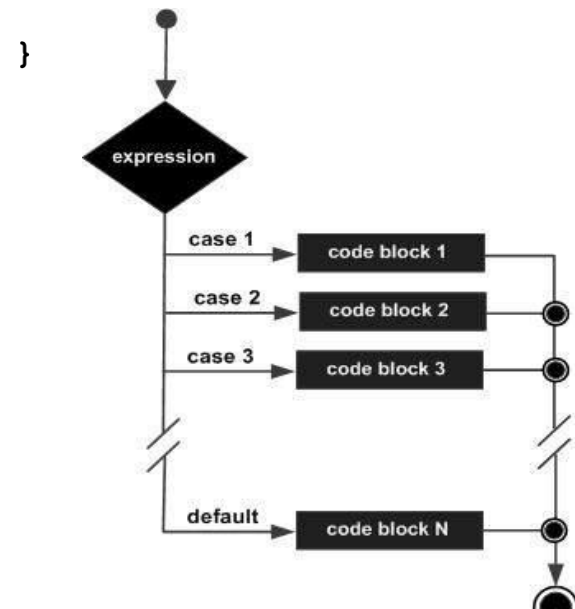
# Practice

- Write a program that find the area of a basic geometric shape
  - Possible shapes: Rectangle, Square, Circle
  - Area
    - Rectangle:  $w * l$
    - Circle:  $\pi * r^2$
    - Square:  $x^2$
- Design a method for calculating area of each shape
- Use if statement in your Main method to call true method depend on the user request. You will ask user for the parameters you need in each function.
- Code: `area.cs`

# Switch statement

- Another tool for making decision in C# is switch statement.
- The general structure of switch statement is show in c# syntax
- The expression in switch statement is evaluated.
- There are different cases that would be compared with the expression value.
- The code block for the case that exactly match with the expression value would be executed.
- If there is no exact match, then the default code block would be selected

```
switch ( expression ){
    case value1 :
        statement-list1
    case value2 :
        statement-list2
    case value3 :
        statement-list3
    case ...
}
```



# Switch statement

- You could implement your calculator with switch-case statement as well similar to the bellow code snippet

```
switch (theOperator)
{
    case "+":
        //ADD UP HERE
        break;
    case "-":
        //SUBTRACT HERE
        break;
    case "*":
        //MULTIPLY HERE
        break;
    case "/":
        //DIVIDE HERE
        break;
    default:
        //DEFAULT CODE HERE
        break;
}
```

# Practice

- Write a C# method that receive the **name** of month and return the number of days we have in that month.
- Call this method from Main method to find the number of days for the month user defined.
- Jan (31), Feb(28), Mar(31), Apr(30), May(31), Jun(30), Jul(31), Aug(31), Sep(30), Oct(31), Nov(30), Dec(31).



- Code: month.cs

# Practice

- Write a C# program that receive the **name** of the month and day of your birthday and return the number of days from the beginning of year to your birthday.
- Using switch-case statement to switch to a specific month if you need any calculation or info for each month and put info/calc in corresponding code block.
- Jan (31), Feb(28), Mar(31), Apr(30), May(31), Jun(30), Jul(31), Aug(31), Sep(30), Oct(31), Nov(30), Dec(31).

- Code: [birthday.cs](#)



# Switch-case or If-else statement?

- Switch case is suitable for the case that we have countable and discrete possible cases.
- If statement is more general for making any conditional statement with complex relational statements or any combination of relations.
- Sample of cases that if statement is better or the only way:
  - If you are working with large number of numbers and you have class of them that can be detected by a conditional statement
    - e.g. you want to do an action for odd numbers and another action for even
  - If you faced with the intervals in most cases if-else should/is better to be used.
    - e.g. grade letter correspond to score code we had in this session



# BREAK ...

---



# Practice: Ordering your pizza

- Your friend is working in a fast food and he/she asked you to write a program for ordering pizza by customers.
- Your program gives customer different option and return the final price that should be paid(price +HST).
- You have different options for a pizza:
  - Type: Veggie(Basic(+\$0)/Mediterian(+\$1)), Pepperoni(Basic(+\$0)/Special(+\$2)), Italian(Basic(+\$0)/Super(+\$2))
  - Size: Small(+\$0), Medium(+\$2), Large(+\$4), Family(+\$5)
  - Cheese type: Mozzarella, Feta(+\$2), Cheddar
  - Additional topping: GroundBeef(+\$3), Hawaiian(+\$2), BBQChicken(+\$2), Corn(+\$1), Spinach(+\$1), Broce(+\$1)
- The base price is \$8 and all the above ones are additional
- You may ask customer for optional TIP 😊
- Code: pizza.cs



## A collage of various numbers in different colors and sizes, representing the concept of numbers. The numbers are scattered across the page, with some being very large and prominent, while others are smaller and less noticeable. The colors include red, green, blue, yellow, and black. The numbers are arranged in a way that they appear to be floating or falling, creating a sense of movement and dynamism. The overall effect is a visual representation of the concept of numbers, which is the central theme of the lesson.

- Write a C# program that generate a random number between 1-100 and let the user should guess that one in 5 steps.
- For random number generation [search class Random in the Internet](#) for MSDN documentation or other web pages you may find helpful (stackoverflow, ...)
- You will not show the random number.
- Any time you ask user for the new guess,
- If the guess is true you can show congratulation message. Otherwise you can [hint user that the random number is greater/smaller](#) than the guess.
- If user couldn't guess, you just show the number at the end.
- [Code: randomGuess.cs \(we see next session\)](#)

## Practice: Guess the random number

- Add difficulty level to your random guess program.
- In this case you can consider 4 different difficulty
  - Easy: interval [1,10] and 4 steps,
  - Moderate: interval [1, 100] and 6 steps,
  - Hard: Interval [1 100] and 4 steps.



- How many guess we need to find exactly such random numbers???

# PA 1

## PROBLEM

Ultrasound imaging of blood vessels is difficult because the contrast between blood and surrounding tissue is low at ultrasound frequencies. One solution is to inject a chemical concoction which basically comprises air-filled microspheres that survive for a few minutes in the blood. If the so-called free-resonance frequency of these microspheres is near the frequency of the ultrasound transducer, they can have an apparent scattering cross section three orders of magnitude larger than their actual geometric cross section. This makes them (and thus the blood in which they are suspended) much more visible in the ultrasound image.

According to a 2005 paper<sup>1</sup> by Sassaroli and Hynynen, the free resonant frequency  $f_0$  (in Hertz) of a microsphere of radius  $a_0$  (in metres) can be determined by the equation

$$f_0 = \frac{1}{2\pi} \sqrt{\frac{1}{\rho (a_0)^2} \left( 3\kappa P_0 + (3\kappa - 1) \frac{2\sigma}{a_0} \right)}$$

where  $\rho$  is the blood density (in kilograms per cubic metre),  $\sigma$  is the blood surface tension (in Newtons per metre),  $\kappa$  is the polytropic index (a dimensionless number related to thermodynamic equilibrium which here will be nearly one), and  $P_0$  is the sum of ambient and blood-vessel pressures (in Pascals).

With values  $P_0 = 108600$  Pa,  $\rho = 1.09 \times 10^3$  kg/m<sup>3</sup>,  $\sigma = 0.0643$  N/m,  $\kappa = 1.0$ , and  $a_0 = 1 \times 10^{-6}$  m, the result of the calculation should be  $f_0 = 3.68 \times 10^6$  Hz.

# THE END ...

---

