**Topic 1: Decisions**

Top of Form

Bottom of Form

**Content**

**Reading**

[Reading](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286908_1)

Decisions

* + Microsoft:docs.microsoft.com
    - [if-else](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/if-else)
    - [?: (ternary conditional) operator](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/operators/conditional-operator)
    - [bool](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/builtin-types/bool)
  + GeeksForGeeks: geeksforgeeks.com
    - [if](https://www.geeksforgeeks.org/c-sharp-decision-making-else-else-ladder-nested-switch-nested-switch/#if)
    - [if-else](https://www.geeksforgeeks.org/c-sharp-decision-making-else-else-ladder-nested-switch-nested-switch/#ifelse)
    - [if-else-if](https://www.geeksforgeeks.org/c-sharp-decision-making-else-else-ladder-nested-switch-nested-switch/#ifelseif)
    - [Nested if](https://www.geeksforgeeks.org/c-sharp-decision-making-else-else-ladder-nested-switch-nested-switch/#nested)
  + TutorialsPoint: tutorialspoint.com
    - [if Statement](https://www.tutorialspoint.com/csharp/if_statement_in_csharp.htm)
    - [if..else Statement](https://www.tutorialspoint.com/csharp/if_else_statement_in_csharp.htm)
  + W3Schools: w3schools.com
    - [If ... Else](https://www.w3schools.com/cs/cs_conditions.asp)

**The if statement**

[The if statement](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286908_1)

Every day you make decisions and think in if statements. For instance, if it's raining, then bring your umbrella. If it is cold, then wear a coat. If you're hungry, then eat. If you're hangry, then everyone run.

You may have heard of an app call IFTTT (<https://ifttt.com>) and you may even use it.

 IFTTT stands for **If** **t**his, **t**hen **t**hat and describes the same basic logic.

In C#, an expression, or condition, is boolean variable (true or false) or evaluates to a boolean.

|  |  |
| --- | --- |
| if (expression)  statement; | When you have one statement to execute when expression evaluates to true. |
| if (expression)  {  statement;  statement;  ...  } | When you have multiple statements to execute when expression evaluates to true. |
| if (expression)  {  statement;  } | This is the best practice even with only one statement. If you add another statement to the "then" part of an if statement without curly brackets { }, you may be at risk for a logic error. See the next row. |
| if (expression)  statement;  statement; | Here the second statement is always executed regardless of the truth value of the expression. |

**Relational Operators for expressions**

[Relational Operators for expressions](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286908_1)

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| == | Checks if the values of two operands are equal or not, if yes then condition becomes true. | (A == B) is not true. |
| != | Checks if the values of two operands are equal or not, if values are not equal then condition becomes true. | (A != B) is true. |
| > | Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true. | (A > B) is not true. |
| < | Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true. | (A < B) is true. |
| >= | Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true. | (A >= B) is not true. |
| <= | Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true. | (A <= B) is true. |

\*Here A and be B can be variables, numeric literals or one of each.

**The if-else statement**

[The if-else statement](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286908_1)

In C#, an expression, or condition, is boolean variable (true or false) or evaluates to a boolean.

|  |  |
| --- | --- |
| if (expression)  statement;  else  statement; | When you have one statement to execute when expression evaluates to true and a singular other statement when the expression evaluates to false. |
| if (expression)  {  statement;  statement;  ...  }  else  {  statement;  statement;  ...  } | When you have multiple statements to execute when expression evaluates to true and multiple other statements when the expression evaluates to false. |
| if (expression)  {  statement;  statement;  ...  }  else  {  statement;  statement;  ...  } | This is the best practice even with only one statement. If you add another statement to the else part of an if-else statement without curly brackets{ }, you may be at risk for a logic error. See the next row.  If you add another statement to the "then" part of the if-else statement without curly brackets { }, you may be at risk for a syntax error. See the second row down. |
| if (expression)  statement;  else  statement;  statement; | Here the third statement is always executed regardless of the truth value of the expression. |
| if (expression)  statement;  statement;  else  statement; | This is a syntax error! |

**The if-else-if statement**

[The if-else-if statement](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286908_1)

In C#, an expression, or condition, is boolean variable (true or false) or evaluates to a boolean.

|  |  |
| --- | --- |
| if (expression)  statement;  else if(expression)  statement;  else  statement; | If, else if and an else.  Example: . If it is warm, go for a bike ride, else if it is snowing, then stay home, else go to the gym. |
| if (expression)  {  statement;  statement;  ...  }  else if (expression)  {  statement;  statement;  ...  }  else if (statement)  {  statement;  statement;  ...  }  else  {  statement;  statement;  ...  } | Keep going! You can leave off the else if your logic does not call for it. Same applies here as if and if-else syntax and logic errors to avoid. |

**Logical Operators**

[Logical Operators](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286908_1)

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| && | Called Logical AND operator. If both the operands are non zero then condition becomes true. | (A && B) is false. |
| || | Called Logical OR Operator. If any of the two operands is non zero then condition becomes true. | (A || B) is true. |
| ! | Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false. | !(A && B) is true. |

\*Here A and B are boolean expressions.

For example, the values of x is between 10 and 20, not including 10 or 20. First we make 2 conditions, A is x is greater than 10 (x > 10) and B is x is less than 20 (x < 20). Finally, the expression A && B is (x > 10) && (x < 20).

**C# if-else statement**

[C# if-else statement](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286908_1)

https://youtu.be/ENJl-UYyUDQ

using System;

namespace BasicIf

{

class Program

{

static void Main(string[] args)

{

// declare bool

bool hungry;

bool haveCash = false; // initaize to false

string answer;

// prompt user

Console.WriteLine("Are you hungry (y/n)? ");

answer = Console.ReadLine();

if (answer.StartsWith('y'))

{

hungry = true;

}

else

{

hungry = false; // not needed if it had been initialized to false

}

if (hungry)

{

// prompt user

Console.WriteLine("Do you have cash (y/n)? ");

answer = Console.ReadLine();

if (answer.StartsWith('y'))

{

haveCash = true;

}

if (!haveCash)

{

Console.WriteLine("Find something at home");

}

else

{

Console.WriteLine("Order some delivery");

}

}

else

{

Console.WriteLine("Do no eat.");

}

}

}

}

**Nested if**

[Nested if](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286908_1)

You can even nest if statements. Here the spacing helps explain the logic at a glance:

if (expression)

{

statement;

...

if (expression)

{

statement;

}

else if (expression)

{

statement;

statement;

...

}

}

else if (expression)

{

statement;

...

if (expression)

{

statement;

}

else if (expression)

{

statement;

statement;

...

}

}

else

{

statement;

statement;

...

}

Any configuration of the above that solves the logic necessary for the program can be used. This is not the only way to nest if and if-else statements, just a generic example. It is over better to use a switch statement to see the logic better and for code maintenance. More on switch coming soon!

**The ternary conditional operator ?:**

[The ternary conditional operator ?:](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286908_1)

In C# there is an elegant shortcut for the if statment, called the ternary conditional operator. It uses the question mark (?) and colon (:). It is best explained with an exmaple:

bool voted = true;

Console.WriteLine(voted ? "voted" : "Did not vote"); // output: Voted

// update bool

voted = false;

Console.WriteLine(voted ? "voted" : "Did not vote"); // output: Did not vote

The equivalent if statement:

bool voted = true;

Console.WriteLine(voted ? "voted" : "Did not vote"); // output: Voted

// update bool

voted = false;

Console.WriteLine(voted ? "voted" : "Did not vote"); // output: Did not vote

The equivalent if statement:

bool voted = true;

if (voted)

{

Console.WriteLine("voted");

}

else

{

Console.WriteLine("Did not vote");

}

// output of above if: Did not vote

// update bool

voted = false;

if (voted)

{

Console.WriteLine("voted");

}

else

{

Console.WriteLine("Did not vote");

}

// output of above if: Did not vote

Clearly, the elegance is in not repeating code and having one line instead of multiple.

Notes:

* + You do not need to write  ~~if (voted == true)~~, simply if (voted) is needed and preferred for the bool type.
  + Similarly, ~~if (voted == false)~~ should be written as if (!voted).

**Decisions: If statements Practice**

[Decisions: If statements Practice](https://dmacc.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_102593_1&content_id=_7286908_1)

Children learn as they grow. They are able to do more household chores as they mature. Use if-else in a Console App that prompts for user input of a child's age, sends it to a method ChoresByAge() that creates a string of chores they are able to do.

For your method, accept the child's age as a parameter and return a string that lists possible chores. Print that string in the Main method. In the Main, ask for the child's age performing input validation. If the user submits non-numeric or invalid (negative age) input, exit Main with the appropriate error message. For non-numeric or non-integer number, use try/catch and for negative number, check with if statement.

* + Age below 2
    - Too young for chores
  + Ages 2-3
    - Pick up toys and books
    - Put garbage in trash can
    - Put dirty laundry in hamper
    - Put laundry in drawer
  + Ages 4-6
    - All previous chores
    - Take care of pets
    - Set and clear the table
    - Make bed
  + Ages 7-11
    - All previous chores
    - Fold laundry
    - Sweep and vacuum
    - Take out trash
    - Meal prep
  + Ages 12+
    - All previous chores
    - Babysit siblings
    - Wash windows
    - Laundry
    - Mow the lawn

To get you started.

class Program

{

private static string ChoresByAge(int ageOfChild)

{

string choreList = "";

// Add if logic to build string

return choreList;

}

static void Main(string[] args)

{

int childAge;

string choresList;

// Get child's from user input

// call method

choresList = ChoresByAge(childAge);

// print choresList

}

}

Try it before checking out the possible solution: [Decisions.cs](https://dmacc.blackboard.com/bbcswebdav/pid-7286942-dt-content-rid-101466146_1/xid-101466146_1)