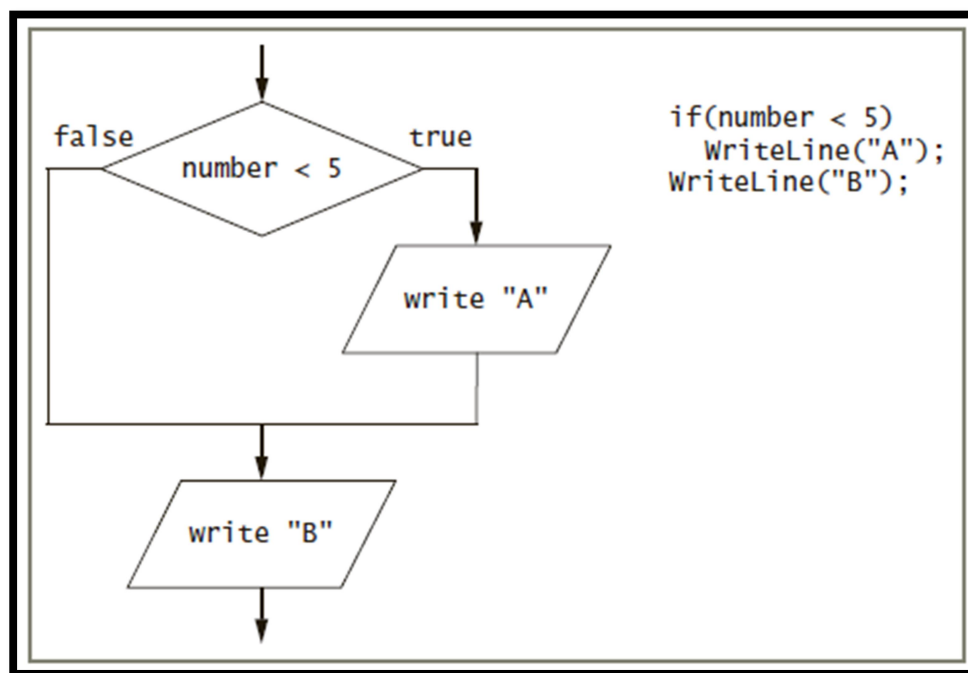


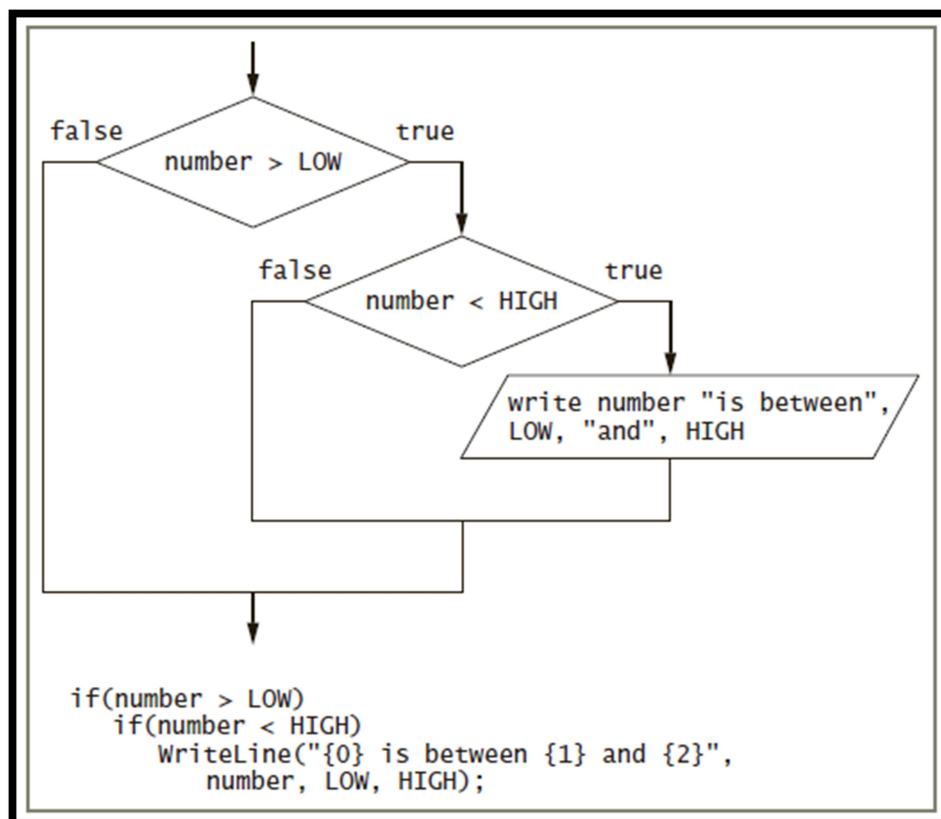
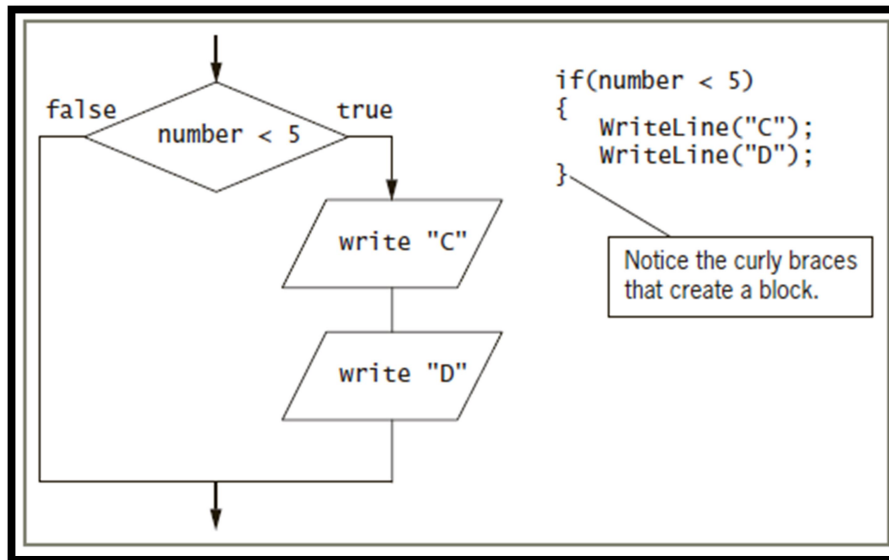
Decision Making (Conditional Statements)

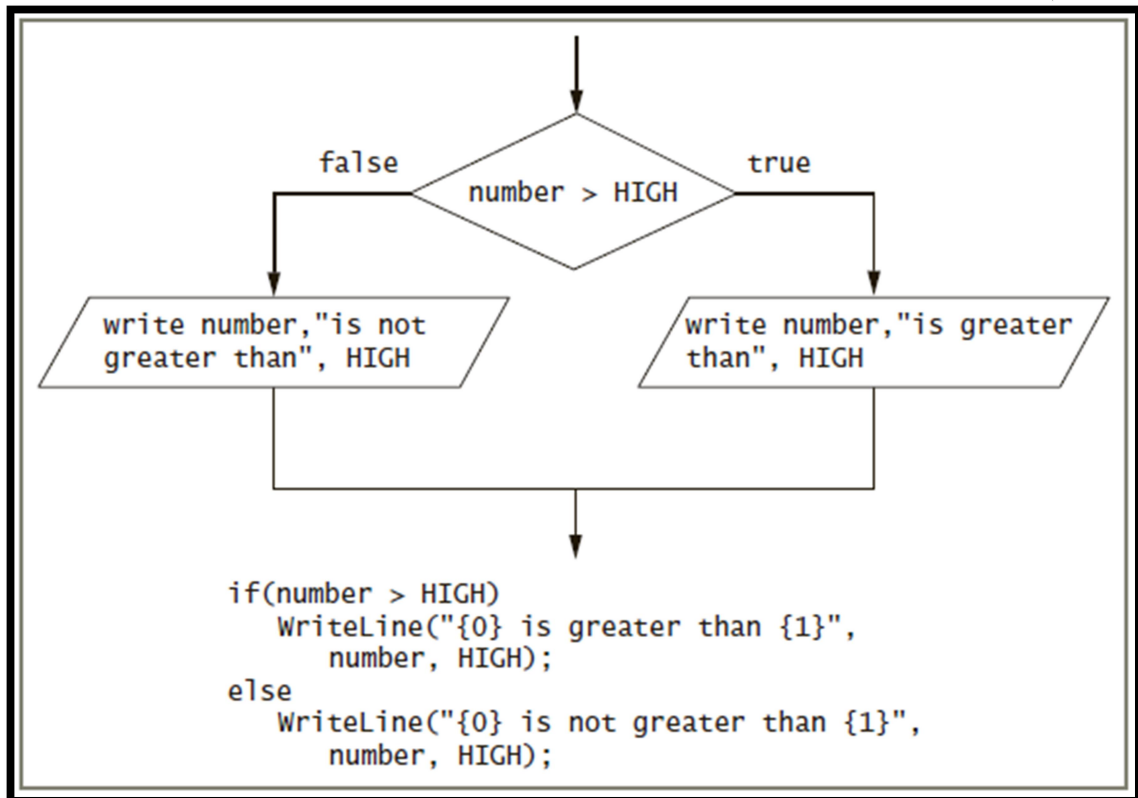
A certain block of code needs to be executed when some condition is fulfilled. A programming language uses control statements to control the flow of execution of program based on certain conditions. The *if* and *if-else* statements are the two most commonly used decision-making statements in C#. You use an if statement to make a single-alternative decision. In other words, you use an if statement to determine whether an action will occur. The if statement takes the following form:

```
if(condition)
{
    // code to be executed
}
```

*Depending upon a decision,
different parts of the program
are executed*







```
int color = "red";
if (color == "red")
    Console.WriteLine("tomato");
else if (color == "yellow")
    Console.WriteLine("banana");
Console.WriteLine("lemon");
```

```
int color = "red";
if (color == "red")
{
    Console.WriteLine("tomato");
}
else if (color == "yellow")
{
    Console.WriteLine("banana");
    Console.WriteLine("lemon");
}
```

Example: The Larger Number

Write a program that reads from the console **two integers** and prints the **larger** of them. Print "Greater number: " + the bigger number.

```
Console.WriteLine("Enter two integers:");
int no1 = int.Parse(Console.ReadLine());
int no2 = int.Parse(Console.ReadLine());
if (no1 > no2)
    Console.WriteLine("Greater number: " + no1);
else
    Console.WriteLine("Greater number: " + no2);
```

Variable Scope

Each variable has a range in which it exists, called **variable scope**. This range specifies where a variable can be used and how long is its **lifetime**. In the C# language, the scope in which a variable exists, starts from the line in which we **defined it** and ends with the first closing curly bracket } (of the method, the if statement, etc.). Thus, it is important to know that **any variable defined inside the body of certain if statement will not be available outside of it**, unless we have defined it higher in the code.

```
int myMoney = 500;
int payDayDate = 10;
int todayDate = 10;
if (todayDate >= payDayDate)
{
    int salary = 5000;
    myMoney = myMoney + salary;
}
Console.WriteLine(myMoney);
Console.WriteLine(salary);
```

The name 'salary' does not exist in the current context

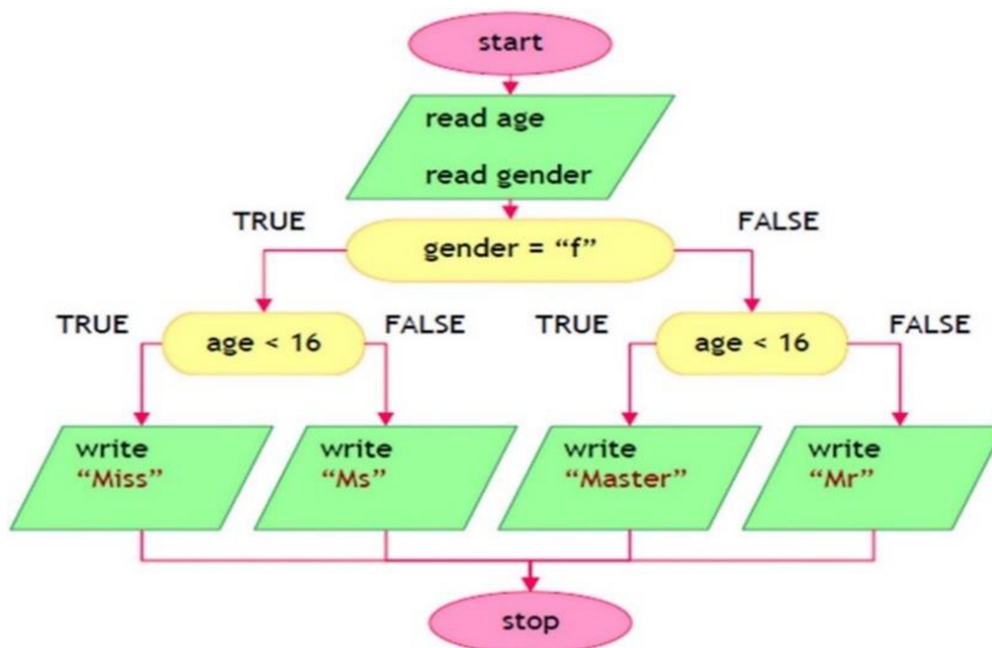
Conditional statements can be nested, i.e. we can put if-else inside another if-else statement. Conditions in the if constructions can be complex, e.g. use logical "AND" or logical "OR".

Example:

Depending on age (decimal number and gender (m / f), print a personal title:

- “**Mr.**” – a man (gender “m”) – 16 or more years old.
- “**Master**” – a boy (gender “m”) under 16 years.
- “**Ms.**” – a woman (gender “f”) – 16 or more years old.
- “**Miss**” – a girl (gender “f”) under 16 years.

Input	Output	Input	Output	Input	Output	Input	Output
12 f	Miss	17 m	Mr.	25 f	Ms.	13.5 m	Master



```
int age = int.Parse(Console.ReadLine());
string gender = Console.ReadLine();
if (age < 16)
{
    if (gender == "m")
        Console.WriteLine("Master");
    else if (gender == "f")
        Console.WriteLine("Miss");
}
else
{
    if (gender == "m")
        Console.WriteLine("Mr.");
    else if (gender == "f")
        Console.WriteLine("Ms.");
}
```

Compound conditionals are conditionals that combine two or more boolean expressions. **Logical operators** are used to combine the expressions (&&, ||, !)

Example:

Checks whether point {x, y} is placed inside the rectangle {x1, y1} – {x2, y2}. The input data is read from the console and consists of 6 lines: the decimal numbers x1, y1, x2, y2, x and y (as it is guaranteed that x1 < x2 and y1 < y2).

Sample Input and Output

Input	Output	Visualization
2 -3 12 3 8 -1	Inside	

```
double x1 = double.Parse(Console.ReadLine());
double y1 = double.Parse(Console.ReadLine());
double x2 = double.Parse(Console.ReadLine());
double y2 = double.Parse(Console.ReadLine());

double x = double.Parse(Console.ReadLine());
double y = double.Parse(Console.ReadLine());

if (x >= x1 && x <= x2 && y >= y1 && y <= y2)
{
    Console.WriteLine("Inside");
}
else
{
    Console.WriteLine("Outside");
}
```

Example:

Check whether a given product is a fruit or a vegetable

- The "fruits" are banana, apple, kiwi, cherry, lemon and grapes.
- The "vegetables" are tomato, cucumber, pepper and carrot.
- Everything else is "unknown".

```
string s = Console.ReadLine();
if (s == "banana" || s == "apple" || s == "kiwi" ||
    s == "cherry" || s == "lemon" || s == "grapes")
    Console.WriteLine("fruit");
else if (s == "tomato" || s == "cucumber"
        || s == "pepper" || s == "carrot")
    Console.WriteLine("vegetable");
else
    Console.WriteLine("unknown");
```

Exercises

- Write a program that inputs a password (one line with random text) and checks if the input matches the phrase "s3cr3t!P@ssw0rd". If it matches, print "Welcome", otherwise "Wrong password!".

- Write a program that inputs an integer in the range [0 ... 9] and outputs it in words in English. If the number is out of range, print "number too big".
- Write a program that inputs 3 numbers and prints whether they are the same (yes / no).
- Write a program that inputs an integer and checks if it is below 100, between 100 and 200 or over 200. Print the appropriate message as in the examples below.

Input	Output
95	Less than 100

Input	Output
120	Between 100 and 200

Input	Output
210	Greater than 200

- We have an integer – the number of points. Bonus score are charged on it, according to the rules described below. Write a program that calculates bonus score for this figure and total points with bonuses.
 - If the number is up to 100 including, bonus score is 5.
 - If the number is larger than 100, bonus score is 20% of the number.
 - If the number is larger than 1000, bonus score is 10% of the number.
 - Additional bonus score (accrued separately from the previous ones):
 - for even number $\rightarrow + 1$ p.
 - o for number, that ends with 5 $\rightarrow + 2$ p.