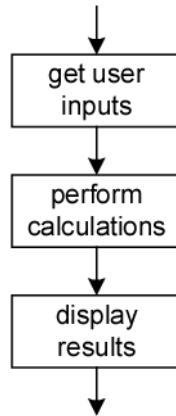


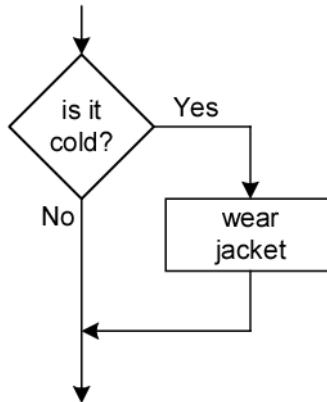
Making Decisions

1. Code Structures

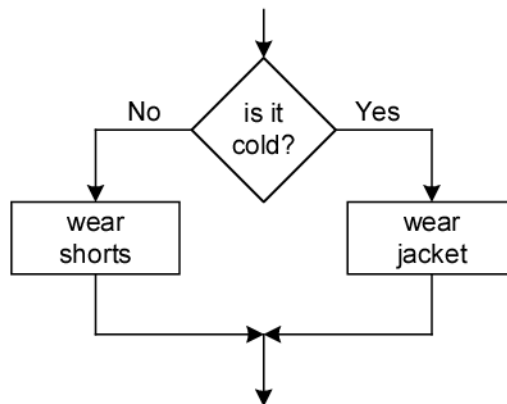
So far all of our programs follow a *sequence structure* because the statements are executed in sequence, without branching off in another direction.



Programs often need to make decisions and to execute some statements only under certain conditions. This can be accomplished with a *decision structure*.



Another version of the *decision structure*:



2. Relational Operators

Relational operators allow the computer to compare numeric and char values and determine whether one is greater than, less than, equal to, or not equal to another.

The relational operators are:

==	(double equal signs) equal
>	greater than
<	less than
>=	greater than or equal
<=	less than or equal
!=	not equal
	(double vertical bars) logical or. You can also use the keyword or
&&	logical and. You can also use the keyword and

The result from evaluating a relational expression, also known as a *Boolean expression*, can only have a *true* or *false* value. For example, if x is greater than y, then the expression $x > y$ will be true, while the expression $y == x$ will be false.

3. If statement

We use the **if** command to get the computer to make a decision. The result of the decision is always going to be either *true* or *false*.

The **if** command is used to execute a statement or block of statements only if the given condition is true. Its syntax is:

```
if (condition)
    statement;
```

where condition is the relational expression that is being evaluated. If the condition is true, the statement is executed. If it is false, the statement is ignored (not executed) and the program continues right after this conditional structure. The statement can be replaced with a block of statements using the braces { } like this

```
if (condition) {
    statement1;
    statement2;
    etc.
}
```

If the condition is true then all of the statements inside the braces are executed in the order given. If the condition is false then the entire block is skipped.

We can additionally specify what we want to happen if the condition is false by using the keyword **else**. Its syntax used in conjunction with if is

```
if(condition)
    statement1;
else
    statement2;
```

In this case, if the condition is true then statement1 is executed, otherwise (i.e. the condition is false) statement2 is executed. Only either statement1 or statement2 is executed, never both. Again, statement1 and statement2 can be replaced with a block of statements enclosed in braces.

```
if(condition) {
    statement3;
    statement4;
    etc.
} else {
    statement5;
    statement6;
    etc.
}
```

Example

Print out whether an input number is less than 50 or not:

```
int number;

cout << "Enter a number between 1 and 100?";
cin >> number;

if (number < 50) {
    cout << "Your number is less than 50\n";
} else {
    cout << "Your number is greater than or equal to 50\n";
}
```

Version two of the same problem:

```
int number;

cout << "Enter a number between 1 and 100?";
cin >> number;

if (50 > number) {
    cout << "Your number is less than 50\n";
} else {
    cout << "Your number is greater than or equal to 50\n";
}
```

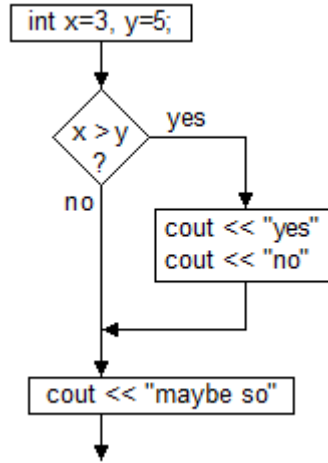
Which version do you prefer?

4. Questions

Match each of the following program segments with the flowchart on the right, and say what they print out.

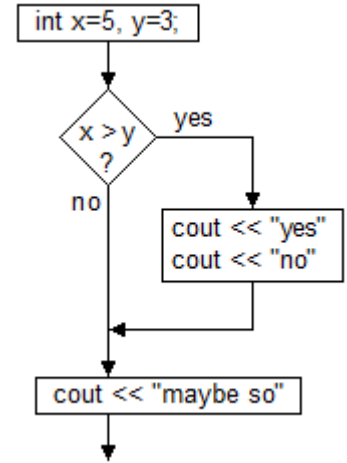
1.

```
int x = 3, y = 5;
if (x > y)
    cout << "yes";
    cout << "no";
    cout << "maybe so";
```



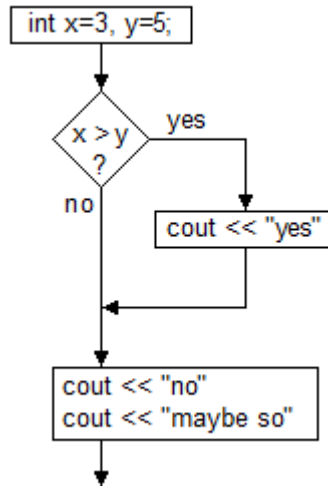
2.

```
int x = 5, y = 3;
if (x > y)
    cout << "yes";
    cout << "no";
    cout << "maybe so";
```



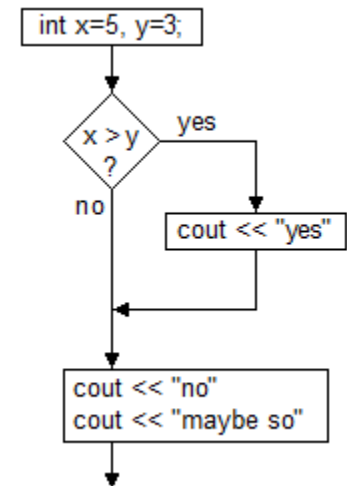
3.

```
int x = 3, y = 5;
if (x > y) {
    cout << "yes";
    cout << "no";
}
cout << "maybe so";
```



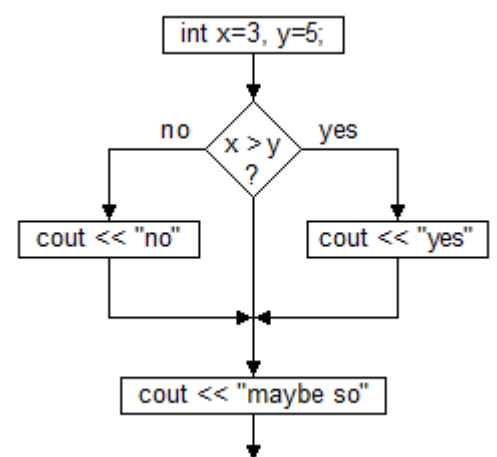
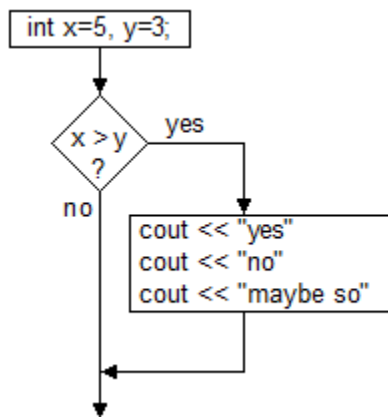
4.

```
int x = 5, y = 3;
if (x > y) {
    cout << "yes";
    cout << "no";
}
cout << "maybe so";
```



5.

```
int x = 3, y = 5;
if (x > y) {
    cout << "yes";
} else {
    cout << "no";
}
cout << "maybe so";
```



6.

```
int x = 3, y = 5;
if (x > y)
    cout << "yes";
else
    cout << "no";
cout << "maybe so";
```

7. I want to find out whether a new number is greater than my current number. (Assume that the two numbers are not equal.) Are the following two questions the same?

- a) Is new number greater than current number?
- b) Is current number less than new number?

Given the above problem statement, which question do you think is easier to understand?

Which one of the following most closely matches the question "Is new number greater than current number?" Which ones are wrong?

- a) if (current_number > new_number)
- b) if (current_number < new_number)
- c) if (new_number > current_number)
- d) if (new_number < current_number)

5. *Nested-If*

When an if statement is inside another if statement.

8.

```
int x = 3, y = 5, z = 7;
if (x > y) {
    if (y > z) {
        cout << "yes";
    } else {
        cout << "no";
    } else {
        cout << "maybe so";
    }
}
```

9.

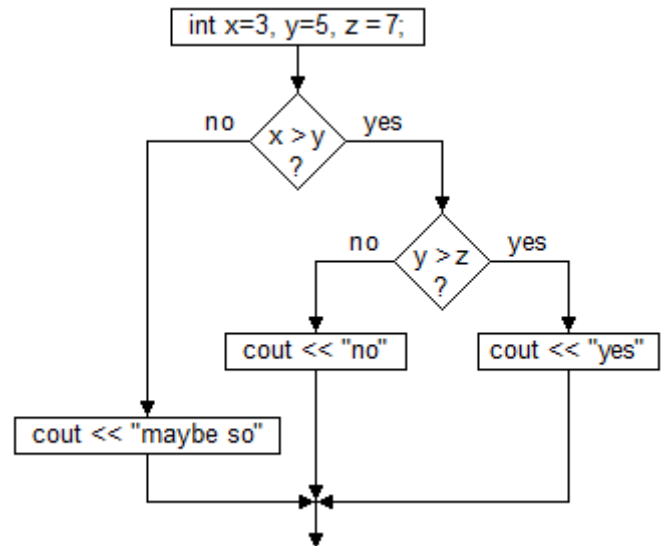
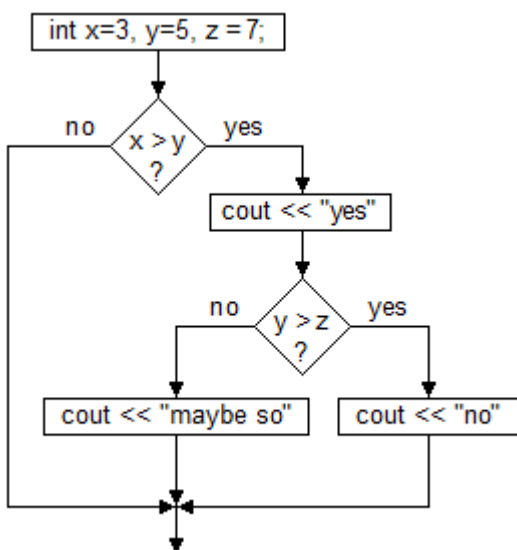
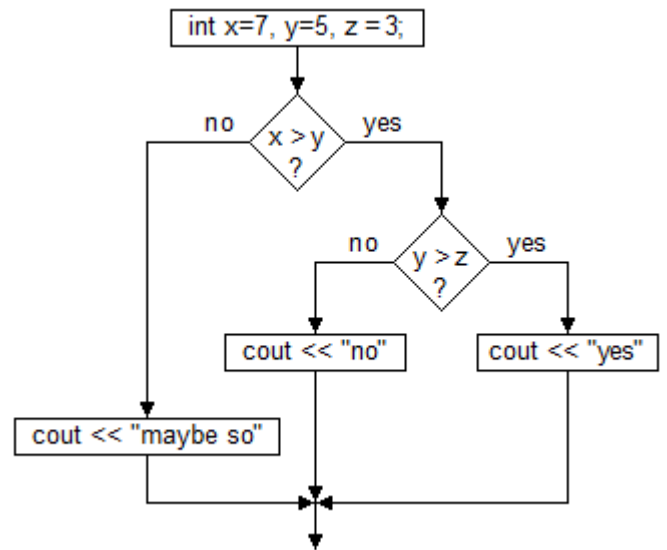
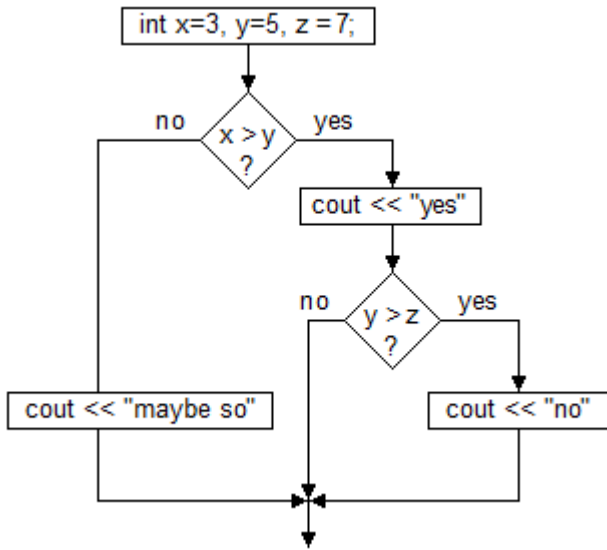
```
int x = 7, y = 5, z = 3;
if (x > y) {
    if (y > z) {
        cout << "yes";
    } else {
        cout << "no";
    } else {
        cout << "maybe so";
    }
}
```

10.

```
int x = 3, y = 5, z = 7;
if (x > y) {
    cout << "yes";
    if (y > z) {
        cout << "no";
    } else {
        cout << "maybe so";
    }
}
```

11.

```
int x = 3, y = 5, z = 7;
if (x > y) {
    cout << "yes";
    if (y > z) {
        cout << "no";
    }
} else {
    cout << "maybe so";
}
```



Instead of

```

if (...) {
} else {
    if (...) {

```

most programmers will do

```

if (...) {
} else if (...) {

```

6. Exercises (Problems with an asterisk are more difficult)

1. Write a program to enter two numbers, *num1* and *num2*. *num1* is divided by *num2* and the result is displayed. Before the division, *num2* is tested for the value 0. If it is a 0, the division does not take place with an appropriate message.
2. Write a program to enter two numbers and then print out the two numbers in ascending order.
3. Write a program to enter a temperature. If the temperature is less than 20 or greater than 100, then print the message “The temperature is in the danger zone.”
4. Just like 3 but in addition if the temperature is not in the danger zone then print the message “The temperature is fine.”
5. Write a program to enter three numbers representing the length of the three sides of a triangle. Print out whether the three lengths can form a valid triangle or not. Hint: the sum of two sides of a triangle must be greater than the third side. If this is true for all three combinations, then you will have a valid triangle.
6. the sum of two side lengths of a triangle is always greater than the third side. If this is true for all three combinations of added side lengths, then you will have a triangle.
7. To qualify for a certain government grant, a person must have worked more than 5 years and earn at least \$35,000. Write a program to enter the number of years a person has worked and his/her annual income, then print out whether the person qualifies for the grant or not. You need to print out one of the following messages:
“You qualify for the government grant”
“You do not qualify because you need to work for more than 5 years”
“You do not qualify because you need to earn more than \$35,000”
8. * Write a program to generate two random numbers between 0 and 100. Ask the user to enter the answer for the sum of the two numbers. Print out whether the answer entered by the user is correct or not. See the [Random Number](#) document on how to generate random numbers.
9. * Write a program to enter a numeric grade which is an integer between 0 and 100. The program will then print out the corresponding letter grade as follows:

90 to 100	A
80 to 89	B
70 to 79	C
60 to 69	D
50 to 59	F

Hint: need to use multiple ELSE IF statements.

10. * Write a program to enter how many books are being purchased. Depending on the number of books purchased, a customer gets a certain percentage discount as follows:

5 or more books	20%
3 to 4 books	10%
Less than 3 books	5%

Print out the percentage discount that the customer gets.

11. * Write a program that displays the following menu:

```

1. Calculate the area of a circle
2. Calculate the area of a rectangle
3. Calculate the area of a triangle
4. Quit

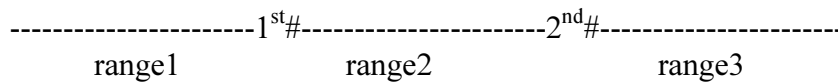
Enter your choice (1-4)?

```

Your program will ask for additional information and perform the appropriate calculation depending on the user choice.

	<u>Formula</u>	<u>Additional user input</u>
The area of a circle is	πr^2 (use 3.14159 for π)	radius
The area of a rectangle is	length * width	length and width
The area of a triangle is	base * height / 2	base and height

12. *** Write a program to enter three numbers and then print out the three numbers in ascending order. Hint 1: First put the first two numbers on a number line. (What I mean by “putting on a number line” is to know which of the two numbers is smaller, just like question 2.) Once you have the first two numbers on the number line then you will only have three different options or three ranges in which the third number can be in.



So you can now test to see which range the third number is in. After knowing the range that the third number is in, you will know the order to print these three numbers out.