Structure of our apps

```
public class ... {
  public static void main(String[] args) {
      ...
  }
}
```

Ingredients of the `main` method

Question

Which "instructions" do we use in the `main` method?

Ingredients of the `main` method

Question

Which "instructions" do we use in the `main` method?

Answer

- declarations

  `type variable;`
- assignments

  `variable = expression;`
- method invocations

  `class.method(arguments);` and

  `object.method(arguments);`

Many problems cannot be solved using only the above "instructions."

Control Structures

CSE 1020

moodle.yorku.ca

Control Structures

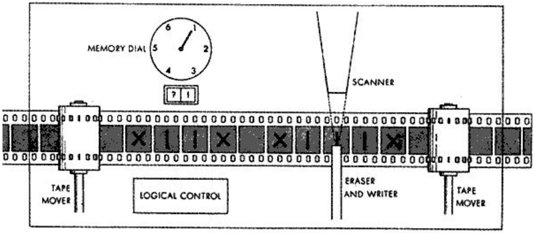
- if statement
- if-else statement
- switch statement
- for statement
- while statement
- do statement

Any of the last three control structures makes Java a so-called Turing complete language.

Turing completeness

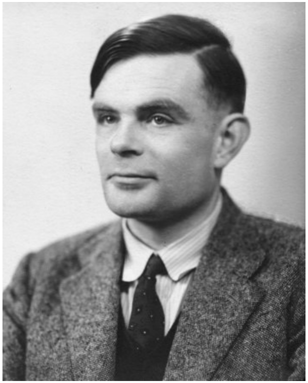
Definition

A programming language is *Turing complete* if a simulator of a Turing machine can written in the programming language.



Alan Turing

Alan Turing (June 23, 1912–June 7, 1954) was an English mathematician. He formalized the notion of computation by means of a machine. This machine was later named the Turing machine. The Turing award, the "Nobel prize of computing" is named after him.


source: ieee.org

Red or Blue

Problem

Prompt the user for input by printing

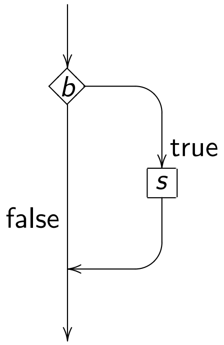
```
0 :  red
1 :  blue
Enter your choice:
```

so that the choice is entered by the user on the same line as the prompt. On the next line, print

`red` *or* `blue`

If statement

```
if (b) {
   s
}
```



If statement

Syntax:

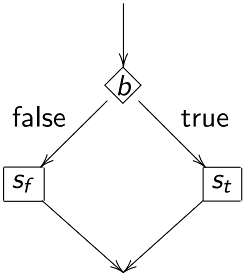
```
if (booleanExpression) {
   statements
}
```

Code conventions:

- `if` should be followed by a single space and
- the body should be indented.

If-else statement

```
if (b) {
   st
} else {
   sf
}
```



If-else statement

Syntax:

```
if (booleanExpression) {
   statements
} else {
   statements
}
```

Code conventions:

- `if` should be followed by a single space and
- the body should be indented.

Scope

Definition

The scope of a variable is that part of the code

- starting from the declaration of the variable,
- ending with the } at level zero.

When we encounter the declaration, we set the level to one.

- Whenever we encounter an {, we increment the level by one.
- Whenever we encounter an }, we decrement the level by one.

Scope

```
output.println("0 : red");
output.println("1 : blue");
output.print("Enter your choice: ");
int choice = input.nextInt();
if (choice == 0) {
   String result = "red";
} else {
   String result = "blue";
}
output.println(result);
```

Scope

```
output.println("0 : red");
output.println("1 : blue");
output.print("Enter your choice: ");
int choice = input.nextInt();
String result;
if (choice == 0) {
   result = "red";
} else {
   result = "blue";
}
output.println(result);
```

Red or Blue

Problem

Prompt the user for input by printing

```
0 :  red
1 :  blue
Enter your choice:
```

so that the choice is entered by the user on the same line as the prompt. Using the class franck.cse1020.Grid, create a grid with one row and one column whose cell has the colour of the given choice.

Red, Blue or Yellow

Problem

Prompt the user for input by printing

```
0 :  red
1 :  blue
2 :  yellow
Enter your choice:
```

so that the choice is entered by the user on the same line as the prompt. On the next line, print

`red` *or* `blue` *or* `yellow`

Many Colours

Problem

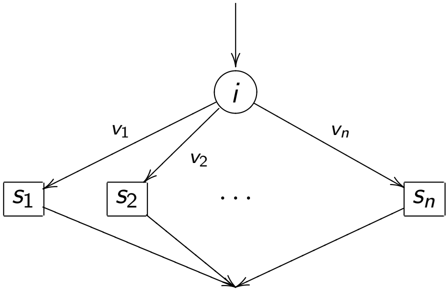
Prompt the user for input by printing

```
0 :  red
1 :  blue
2 :  yellow
3 :  cyan
4 :  magenta
5 :  orange
6 :  pink
Enter your choice:
```

so that the choice is entered by the user on the same line as the prompt. On the next line, print the corresponding colour.

Switch statement

```
switch (i) {
  case v1 : s1
            break;
  case v2 : s2
            break;
  ...
  case vn : sn
            break;
}
```



Switch statement

Syntax:

```
switch (integerExpression) {
   case integerValue:
      statements
      break;
   case integerValue:
      statements
      break;
   ...
   default:
      statements
}
```

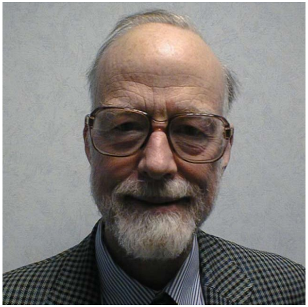
Switch statement

Code conventions:

- `switch` should be followed by a single space,
- `case` should be followed by a single space, and
- the body should be indented.

Tony Hoare

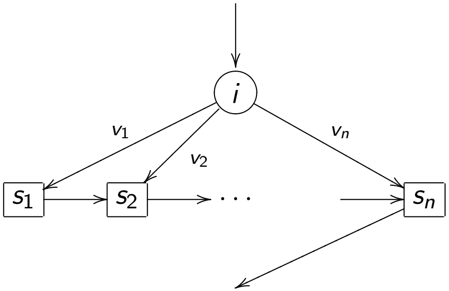
Sir Charles Antony Richard Hoare (born January 11, 1934) is a British computer scientist. He is best known for the development of Quicksort, an algorithm to sort elements. He also proposed the switch statement. In 1980, he received the Turing award.



source: research.microsoft.com

Switch statement without breaks

```
switch (i) {
  case v1 : s1
  case v2 : s2
  ...
  case vn : sn
}
```



Line of stars

Problem

Prompt the user for a non-negative integer

```
    Enter a non-negative integer:
```

so that the integer n is entered by the user on the same line as the prompt. On the next line, print n $*$'s.

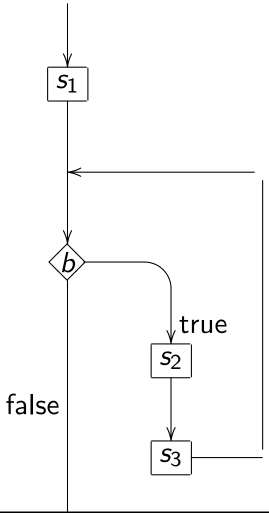
Loops

CSE 1020

moodle.yorku.ca

For statement

```
for(s1; b; s3) {
  s2
}
```



For statement

Syntax

```
for $(s_1;\ b;\ s_3)$ {
    $s_2$;
}
```

Code conventions:

- for should be followed by a space and
- the body should be indented.

Line of stars

Problem

Prompt the user for a non-negative integer

```
    Enter a non-negative integer:
```

so that the integer n is entered by the user on the same line as the prompt. On the next line, print n $*$'s.

Line of blocks

Problem

Prompt the user for a non-negative integer

```
    Enter a non-negative integer:
```

so that the integer c is entered by the user on the same line as the prompt. Using the class franck.cse1020.Grid, create a grid with one row and c columns, every second make a cell of the grid red (going from left to right).

Line of blocks

Exercise

Prompt the user for a non-negative integer

```
    Enter a non-negative integer:
```

so that the integer c is entered by the user on the same line as the prompt. Using the class franck.cse1020.Grid, create a grid with one row and c columns, every second color a cell of the grid, alternating red and black (going from left to right).

Line of numbers

Problem

Prompt the user for a non-negative integer

```
    Enter a non-negative integer:
```

so that the integer n is entered by the user on the same line as the prompt. On the next line, print 1, 2, \ldots $n-1$, n, separated by a single space.

Block of stars

Problem

Prompt the user for two positive integers

```
    Enter the number of rows:
    Enter the number of columns:
```

so that the integers r and c are entered by the user on the same line as the prompts. Print r lines each consisting of c $*$'s.

Block of blocks

Problem

Prompt the user for two positive integers

```
Enter the number of rows:
Enter the number of columns:
```

so that the integers *r* and *c* are entered by the user on the same line as the prompts. Using the class franck.cse1020.Grid, create a grid with *r* rows and *c* columns, every second make a cell of the grid red (going from left to right, and from top to bottom.)

Block of blocks

Exercise

Prompt the user for two positive integers

```
Enter the number of rows:
Enter the number of columns:
```

so that the integers *r* and *c* are entered by the user on the same line as the prompts. Using the class franck.cse1020.Grid, create a grid with *r* rows and *c* columns, every second color a cell of the grid, alternating red and black (going from left to right, and from top to bottom.)

Tree

Problem

Prompt the user for a positive integer

```
    Enter the height of the tree:
```

so that the integer h is entered by the user on the same line as the prompts. Print a tree of height $h+1$. For example, if $h = 4$, print

```
   *
  ***
 *****
*******
   *
```

Tree

Exercise

Prompt the user for a positive integer

```
    Enter the height of the tree:
```

so that the integer h is entered by the user on the same line as the prompts. Print a tree of height $h + 1$ using the class franck.cse1020.Grid.

Print a file

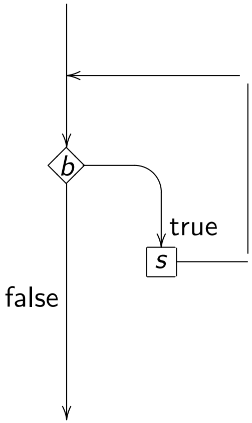
Problem

Prompt the user for a file name

```
    Enter a file name:
```

so that the name is entered by the user on the same line as the prompt. Print the content of the file.

While statement



While statement

Syntax

$\text{while } (b)\ \{$
$\quad s;$
$\}$

Code conventions:

- while should be followed by a space and
- the body should be indented.

For and while loops

Theorem

Every for-loop can be expressed as a while-loop.

Proof.

```
for ($s_1$; $b$; $s_2$) {
    $s_3$;
}
```

can be expressed as

```
{
    $s_1$;
    while ($b$) {
        $s_3$;
        $s_2$;
    }
}
```

□ □

For and while loops

Theorem

Every while-loop can be expressed as a for-loop.

Print a triangle

Problem

Prompt the user for a positive integer

```
    Enter a positive integer:
```

so that the integer n is entered by the user on the same line as the prompts. Print a line with 1 $*$, a line with 2 $*$'s, ..., a line with $n-1$ $*$'s, and a line with n $*$'s.

Reprompt

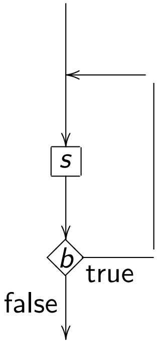
Problem

Prompt the user for a positive integer

```
    Enter a positive integer:
```

so that the integer n is entered by the user on the same line as the prompts. Print a line with 1 $*$, a line with 2 $*$'s, . . . , a line with $n-1$ $*$'s, and a line with n $*$'s. Reprompt the user if they enter a non-positive integer.

Do statement



Do statement

Syntax

```
do {
    s;
} while (b);
```

Code conventions:

- while should be followed by a space and
- the body should be indented.

For and do Loops

Theorem

Every for-loop can be expressed as a do-loop.

Theorem

Every do-loop can be expressed as a for-loop.

Question

So which loop should we use?

For and do Loops

Theorem

Every for-loop can be expressed as a do-loop.

Theorem

Every do-loop can be expressed as a for-loop.

Question

So which loop should we use?

Answer

It is a matter of taste. If you know the number of iterations in advance, a for-loop may be most appropriate. If the loop has to be executed at least once, a do-loop may be most appropriate.

Prime

Exercise

Prompt the user for a positive integer

```
Enter a positive integer:
```

so that the integer n is entered by the user on the same line as the prompt. On the next line, print

n `is prime`

if n is prime and

n `is not prime`

otherwise.

Prime

The New York Times

New Method Said to Solve Key Problem in Math

By Sarah Robinson

Three Indian computer scientists have solved a longstanding mathematics problem by devising a way for a computer to tell quickly and definitively whether a number is prime – that is, whether it is evenly divisible only by itself and 1.

New York Times, August 8, 2002

To do

- Study Chapter 5 of the textbook.
- Complete Check05A from the textbook before February 15.
- Start thinking about your project proposal (due on February 18).