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

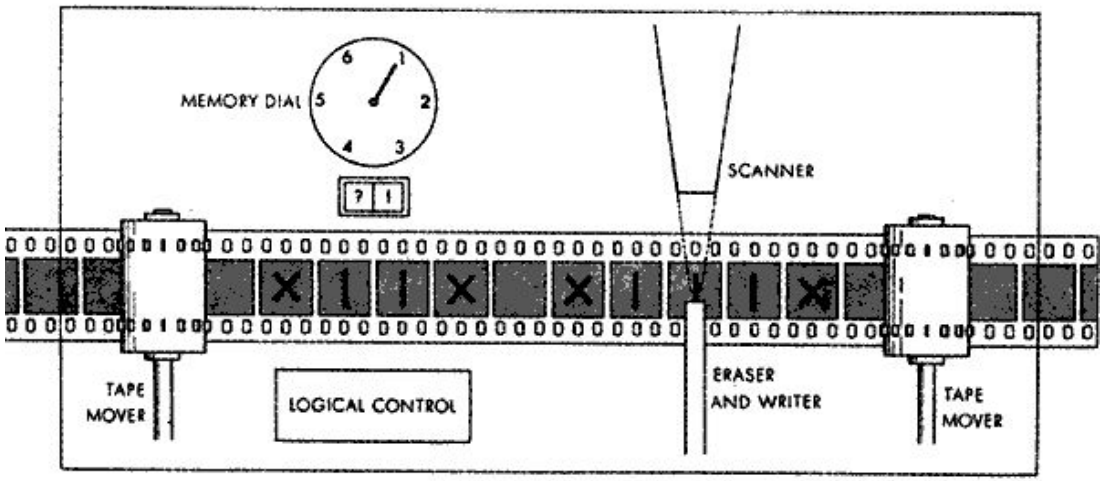
- if statement
- if-else statement
- switch statement
- for statement
- while statement
- do statement

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.



This notion will be covered in more detail in the course "Introduction to the Theory of Computation" (CSE 2001).

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

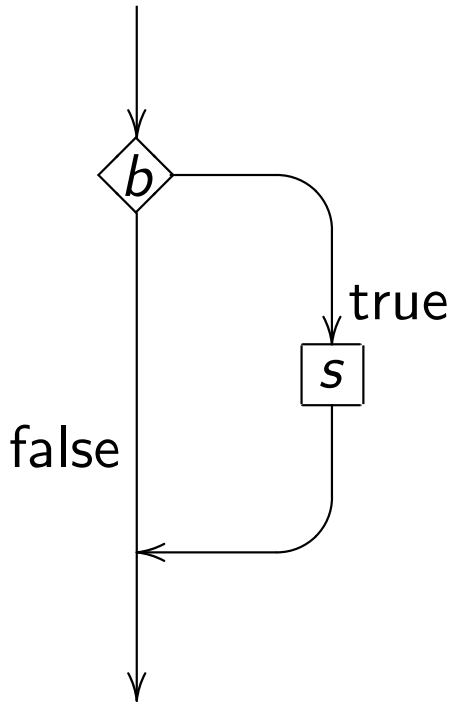
```
0 :  red
1 :  blue
Enter your choice:
```

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:

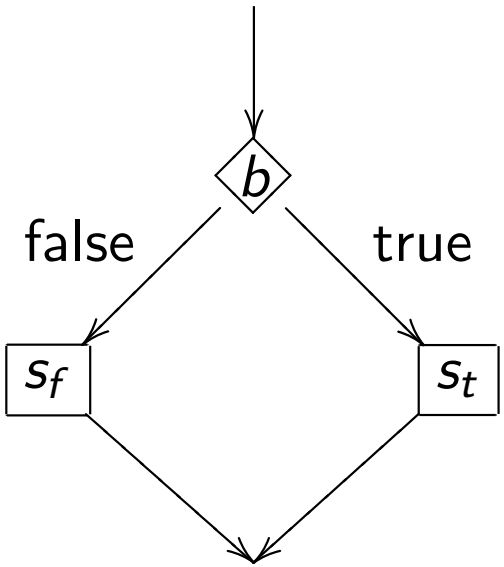
```
if (booleanExpression) {
   statements
}
```

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

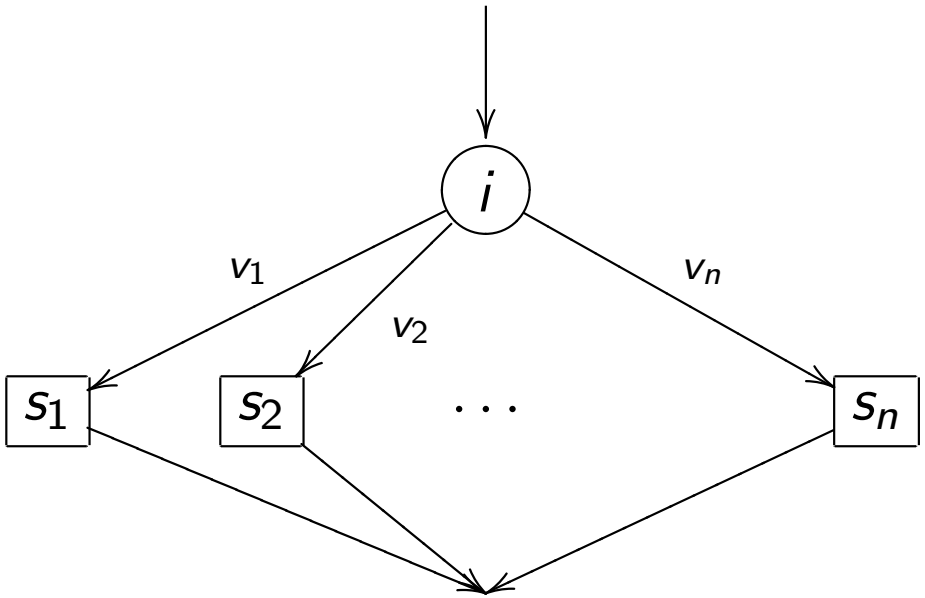
Prompt the user for input by printing

```
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
}
```

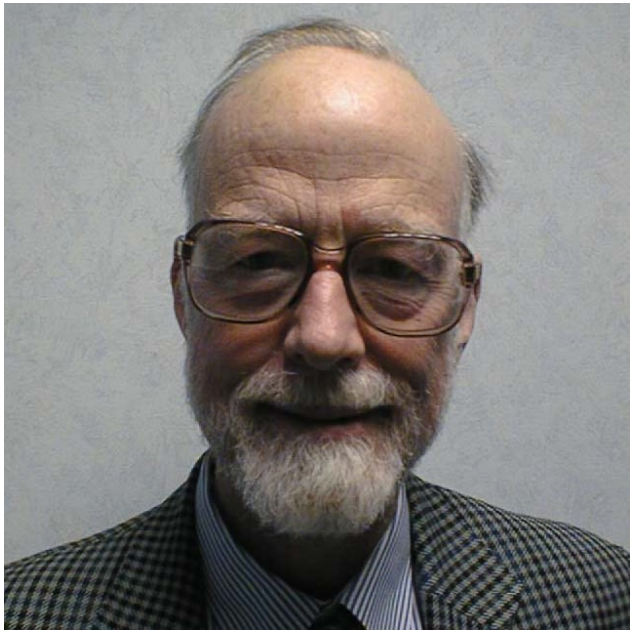
Switch statement

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

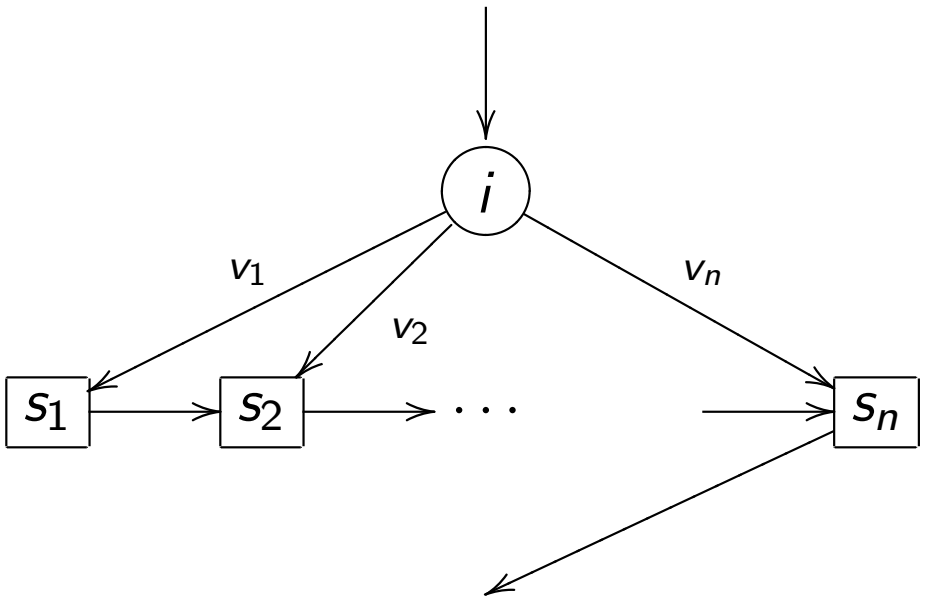
Sir Charles Antony Richard Hoare (born January 11, 1934) is a British computer scientist. He is best known for the development of Quicksort, an algorithm to sort elements. He also proposed the switch statement. In 1980, he received the Turing award.



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.

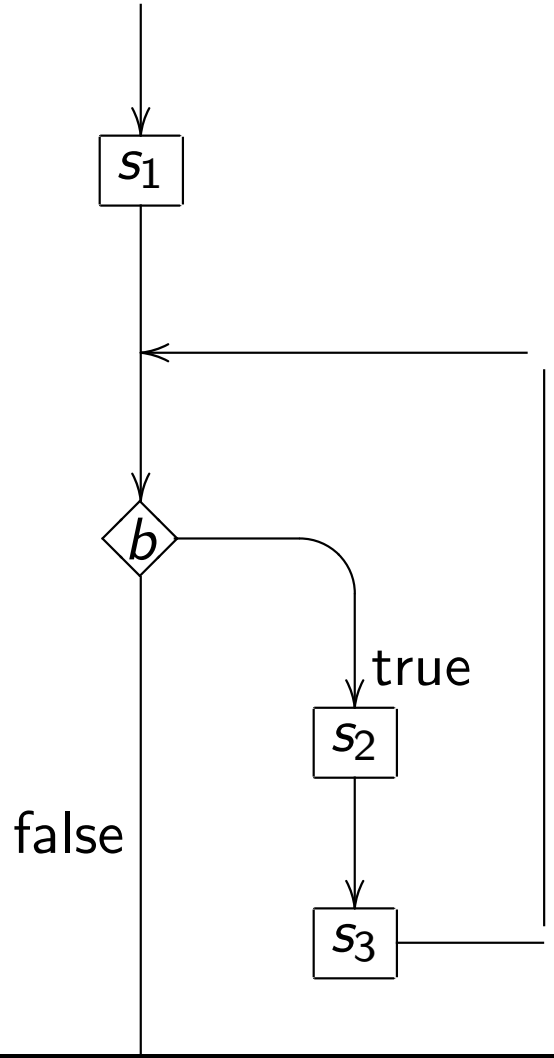
Loops

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.

Loop Invariant

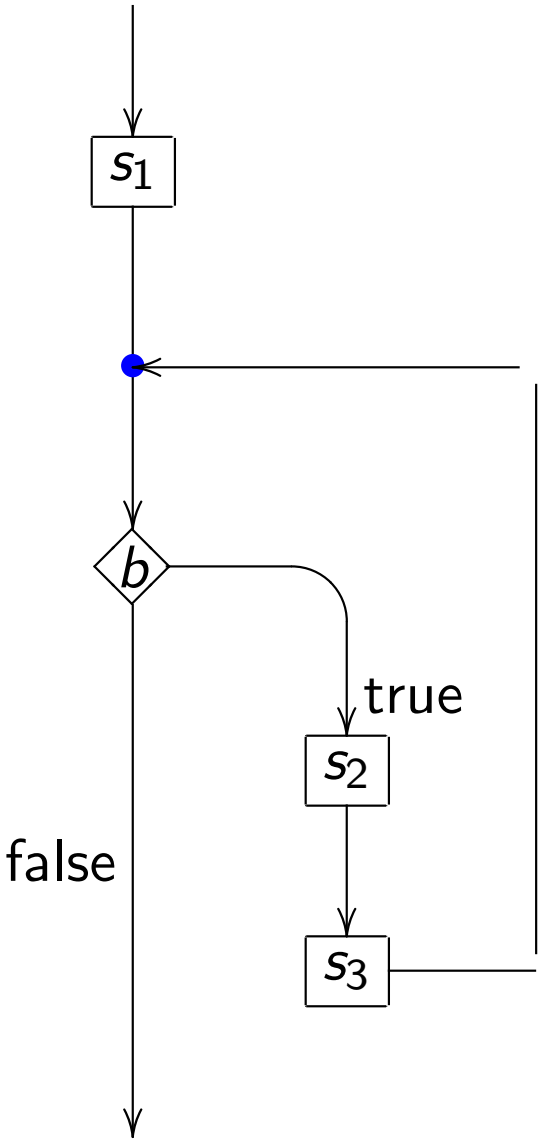
Definition

Given a loop, a boolean expression is a *loop invariant* of the loop if it holds at the beginning of every iteration of the loop.

C.A.R. Hoare. An Axiomatic Basis for Computer Programming. *Communications of the ACM*, 12(10): 576–580, October 1969.

Loop invariant for a for statement

- •: where the loop invariant should hold



Loop Invariant

Consider the loop

```
for (int i = 0; i < n; i++) {
    output.print("*");
}
```

Loop invariants for this loop are

Loop Invariant

Consider the loop

```
for (int i = 0; i < n; i++) {
    output.print("*");
}
```

Loop invariants for this loop are

- true

Loop Invariant

Consider the loop

```
for (int i = 0; i < n; i++) {
    output.print("*");
}
```

Loop invariants for this loop are

- true
- $\texttt{i} \geq 0$

Loop Invariant

Consider the loop

```
for (int i = 0; i < n; i++) {
    output.print("*");
}
```

Loop invariants for this loop are

- true
- `i` ≥ 0
- `i` $\leq n$

Loop Invariant

Consider the loop

```
for (int i = 0; i < n; i++) {
    output.print("*");
}
```

Loop invariants for this loop are

- true
- `i` ≥ 0
- `i` $\leq n$
- i *'s have been printed

Loop Invariant

Consider the loop

```
for (int i = 0; i < n; i++) {
    output.print("*");
}
```

Loop invariants for this loop are

- true
- `i` ≥ 0
- `i` $\leq n$
- i *'s have been printed
- `i` ≥ 0 && `i` $\leq n$ && i *'s have been printed

To do

- Study pages 172–193 of the textbook.