EECS1022
MOBILE COMPUTING

CONTROL STRUCTURES

SELECTION

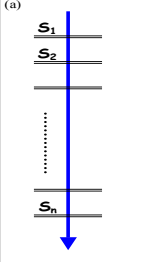(SLIDES ADAPTED FROM PROF.H. ROUMANI)

PROF. Y. LESPÉRANCE

Dept. of Electrical Engineering & Computer Science
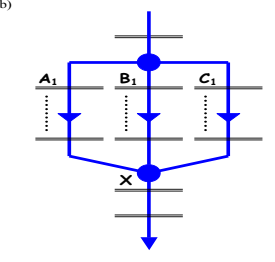
1

FLOW OF CONTROL

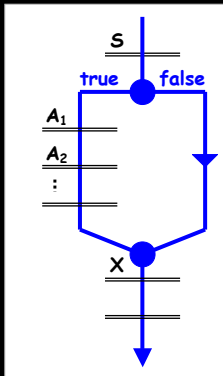- What is control; how does it flow?

- Sequence versus Selection Flow

2

SELECTION: IF


```
Statement-S
if (condition)
{
    Statement-A1
    Statement-A2

       ⋮

}
Statement-X

   ⋮
```

3

EXAMPLE

Write a method that reads an int and out-puts its *abs* value without using `Math.abs`.

```java
public int myAbs(int n)
{
    int result = n;
    if (n < 0)
    {
        result = -n;
    }
    return result;
}
```
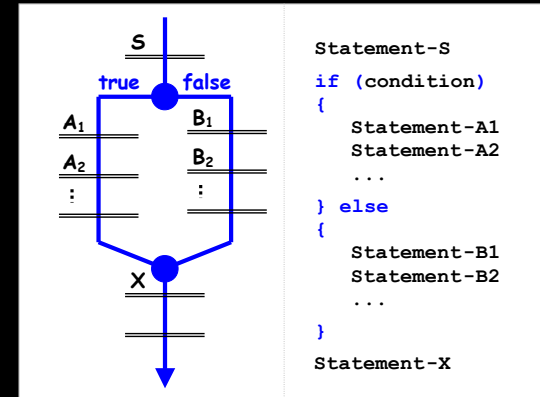
4

PITFALL

What is wrong with this?

```
public int myAbs(int n)
{
    int result = n;
    if (n < 0);
    {
        result = -n;
    }
    return result;
}
```

5

SELECTION: IF-ELSE


```
Statement-S
if (condition)
{
    Statement-A1
    Statement-A2
    ...
} else
{
    Statement-B1
    Statement-B2
    ...
}
Statement-X
```

6

EXAMPLE

Rewrite the **myAbs** body using if-else

```
int result;
if (n < 0)
{
    result = -n;
}
else
{
    result = n;
}
return result;
```

7

PITFALL

What is wrong with this?

```
if (n < 0)
{
    int result = -n;
}
else
{
    int result = n;
}
return result;
```

8

BUILDING THE CONDITION

- **Relational Expression**
  ```
  if (k < 0)
  ```

- **Boolean Variable**
  ```
  boolean b = k < 0
  if (b)
  ```

- **Boolean Expression**
  ```
  if (k < 0 || b && m > h)
  ```
  Uses boolean operators: &&, ||, and !

9

OPERATOR PRECEDENCE

```
. ++ -- !
cast
* / %
+ -
<  <=  >  >=
== !=
&&
||
= op=
```

10

EXAMPLE

- Express the condition x ∈ [a, b)

  ```
  if (x >= a && x < b)
  ```

- Express the condition x ∉ [a, b)

  ```
  if (!(x >= a && x < b))
  ```

  Can use deMorgan's Law to convert negated conjunctions to disjunctions.

  ```
  if (x < a || x >= b)
  ```

11

EXAMPLE

Given two object references x and y, write a condition to determine if the objects they reference are equal.

Which one is correct and why:

- ```
  if (x == y)
  ```
- ```
  if (x.equals(y))
  ```
- ```
  if (x.equals(y) && x != null)
  ```
- ```
  if (x != null && x.equals(y))
  ```

12

EXERCISES

Implement these methods:

- public boolean isOdd(int n)
  Returns true if n is odd else returns false.

- public char getFullGrade(int mark)
  Returns the full letter grade (A,B,C,D,E,F) given the mark (out of 100) in a course.

- public boolean isLeap(int year)
  Returns true if the year is a leap else returns false.
  A leap year is divisible by 4 and if it is divisible by 100 then it must also be divisible by 400; e.g. 2016 is, 2000 is, but 2100 is not.

13

EECS1022
MOBILE COMPUTING
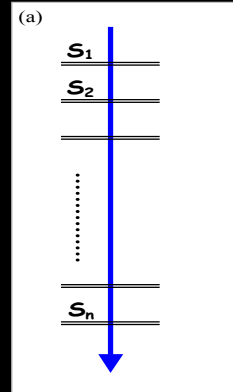
CONTROL STRUCTURES
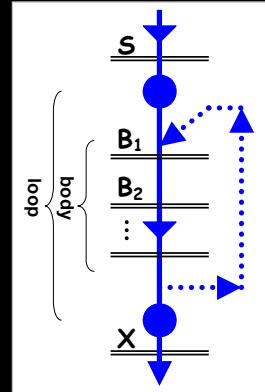# ITERATION
(SLIDES ADAPTED FROM PROF. H. ROUMANI)

PROF. Y. LESPÉRANCE
Dept. of Electrical Engineering & Computer Science

1
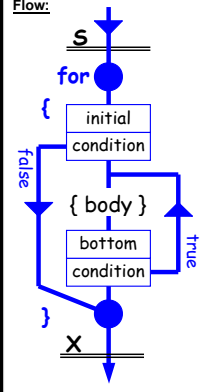
ITERATION FLOW OF CONTROL

2

ITERATION: FOR

Flow:

S
for
{
initial
condition
false
{ body }
bottom
condition
true
}
X

Syntax:

```
Statement-S
for (initial; condition; bottom)
{
    body;
}
Statement-X
```

Algorithm:

1. Start the for scope
2. Execute initial
3. If condition is false go to 9
4. Start the body scope {
5. Execute the body
6. End the body scope }
7. Execute bottom
8. If condition is true go to 4
9. End the for scope

3

EXAMPLE

Output a table of square roots in [0,9].

```
for (int i = 0; i < 10; i = i + 1)
{
    double root = Math.pow(i, 0.5);
    System.out.print(i);
    System.out.print("\t");
    System.out.println(root);
}
```

4

Slide 5

```
for (initial; condition; bottom)
```

```
for (int i = 0; i < MAX; i = i + 1)
{
    ...
}
```

```
int i = 0;
for (; i < MAX; i = i + 1)
{
    ...

}
```

5

Slide 6

```
for (initial; condition; bottom)
```

- Can it be omitted?
- Can it be set to the literal `true`?
- What if it were false at the beginning?
- Is it monitored throughout the body?

6

Slide 7

```
for (initial; condition; bottom)
```

- Can be any statement.
- Can be omitted.
- Will the loop become infinite if it is omitted?

7

Slide 8

EXAMPLE

Write a fragment to output the exponents of all powers of 2 that are smaller than a million.

Correct output:

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
```

```
final int MILLION = 1000000;
for (int expo = 0; Math.pow(2, expo) < MILLION; expo++)
{
    System.out.print(expo);
    System.out.print(" ");
}
System.out.println();
```

8

EXAMPLE

Rewrite the fragment so it only outputs the exponent of the greatest power of 2 that is smaller than a million.

```
int expo = 0;
for (; Math.pow(2, expo) < MILLION; expo++)
{
}
System.out.println(expo - 1);
```

```
int expo = 0;
for (; Math.pow(2, expo) < MILLION; expo++);
System.out.println(expo - 1);
```

9

NESTED LOOPS

- Must be fully nested

- Nested structures imply nested scopes

```
for (int i = 0; i < 10; i++)
{
    for (int j = 0; j < 10; j++)
    {
        System.out.println(i + "  " + j);
    }
}
```

10

EXERCISES

Implement these methods:

- public boolean isPrime(int n)
  Returns true if n is prime, else returns false.

- public int log2 (int n)
  Returns the number of repeated division of n by 2 until the result of the division is 1

- public double factorial(int n)
  Returns n!

- public double oddRecipSum (int n)
  Returns the sum of the reciprocals of odd integers with alternating signs between 1 and 10^6 (should around π/4)

11