Chapter 5

Control Structures

Outline

5.1 Selection
   5.1.1 Flow of Control
   5.1.2 The if Statement
   5.1.3 Building the Condition
   5.1.4 Multway Branching

5.2 Iteration
   5.2.1 Flow of Control
   5.2.2 The for Statement
   5.2.3 Building the Loop
   5.2.4 Nested Loops

5.3 Application
   5.3.1 Exception-Free Input Validation
   5.3.2 File I/O

5.1.1 Flow of Control

• What is control; how does it flow?
• Sequence versus Selection Flow

Diagram showing flow of control and selection flow.
### 5.1.2 The if Statement

#### Code

```java
if (condition) {
    Statement-A1
    Statement-A2
...
} else {
    Statement-X
}
```

#### Example

Write a fragment that reads an int and outputs its abs value without using `Math.abs`.

```java
output.print("Enter an integer ... ");
int entry = input.nextInt();
int absValue = entry;
if (entry < 0) {
    absValue = -entry;
}
output.println(absValue);
```
Pitfall
What is wrong with this?

output.print("Enter an integer ... ");
int entry = input.nextInt();
int absValue = entry;
if (entry < 0);
{
    absValue = -entry;
}
output.println(absValue);

The if-else Statement

Example
Rewrite the abs value fragment using if-else
Example
Rewrite the abs value fragment using if-else

```java
int absValue;
if (entry < 0)
{
    absValue = -entry;
}
else
{
    absValue = entry;
}
```

Pitfall
What is wrong with this?

```java
if (entry < 0)
{
    int absValue = -entry;
}
else
{
    int absValue = entry;
}
```

5.1.3 Building the Condition

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

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

- Boolean Expression
  ```java
  if (k < 0 || b && m > h)
  ```

Uses boolean operators: &&, ||, and !
Operator Precedence

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

Examples

• Express the condition $x \in [a, b)$

• Express the condition $x \notin [a, b)$

Examples

• Express the condition $x \in [a, b)$
  
  if ($x \geq a \&\& x < b$)

• Express the condition $x \notin [a, b)$
  
  if (!(x >= a &\& x < b))
  
  Can use deMorgan’s Law to convert negated conjunctions to disjunctions.
  
  if ($x < a || x \geq b$)
Example
Write a fragment to determine if the objects of two given references \( x \) and \( y \) are equal.

\[
\text{if} \ (x.\text{equals}(y))
\]

But what if \( x \) were null?

Example
Write a fragment to determine if the objects of two given references \( x \) and \( y \) are equal.

\[
\text{if} \ (x \neq \text{null} \ \&\& \ x.\text{equals}(y))
\]

But wouldn't this crash anyway?

Lazy (short-circuit) evaluation of \&\& and ||.

5.1.4 Multiway Branching
Reduce an \( n \)-way branch to \( n-1 \) 2-way branches

\[
\begin{array}{c|c|c}
& S & \text{Statement-1} \\
\text{true} & A_1 & \{ \text{Statement-A1} \\
\text{false} & A_2 & \{ \text{Statement-A2} \\
& \ldots & \} \text{ else if (condition-2)} \\
& B_1 & \{ \text{Statement-B1} \\
& B_2 & \{ \text{Statement-B2} \\
& \ldots & \} \text{ else} \\
& C_1 & \{ \text{Statement-C1} \\
& C_2 & \{ \text{Statement-C2} \\
& \ldots & \} \text{ } \text{Statement-X}
\end{array}
\]
Example

Write a fragment to determine the full letter grade given the mark (out of 100) in a course.

```java
char grade;
if (mark >= 80)
    grade = 'A';
else if (mark >= 70)
    grade = 'B';
else if (mark >= 60)
    grade = 'C';
else if (mark >= 50)
    grade = 'D';
else
    grade = 'F';
```

5.2.1 Flow of Control

[Diagram showing flow of control]

Copyright © 2006 Pearson Education Canada Inc.
5.2.2 The for statement

Flow:

Syntax:

Algorithm:

Example

```java
final int MAX = 10;
final double SQUARE_ROOT = 0.5;
for (int i = 0; i < MAX; i = i + 1)
{
    double sqrt = Math.pow(i, SQUARE_ROOT);
    output.print(i);
    output.print(TEXT.TAB) // tab
    output.println(sqrt);
}
```
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?

for (initial; condition; bottom)

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

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

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

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

As a second example, rewrite the fragment so it only outputs the exponent of the greatest power of 2 that is smaller than a million.

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

5.2.3 Building the Loop

- Sentinel-based example
  Write a program that reads integers with a -1 sentinel and outputs their arithmetic mean.

- Number statistics examples
  Read numbers and determine their largest, smallest, second-largest, …
Write a program that reads integers with a -1 sentinel and outputs their arithmetic mean.

Pseudo-code:
```java
for (?; not sentinel; ?)
{
    process the int
    read an int
}
```

Sentinel-Based Looping

Write a program that reads integers with a -1 sentinel and outputs their arithmetic mean.

Pseudo-code:
```
for (?; not sentinel; ?)
{
    process the int
    read an int
}
```

**Priming needed**

read an int
```
for (?; not sentinel; ?)
{
    process the int
    read an int
}
```

Sentinel-Based Looping

read an int
```
for (read an int; not sentinel; ?)
{
    process the int
    read an int
}
```

Copyright © 2006 Pearson Education Canada Inc.
Sentinel-Based Looping

```java
for (read an int; not sentinel; ?)
{
    process the int
    read an int
}
```

```java
for (read an int; not sentinel; read an int)
{
    process the int
}
```

- How do you count the entries?
- How do you compute the mean?
- Is a cast needed?

---

Number Statistics

- Finding the max entry
- Using and challenging a candidate
- Seeding the candidate
- A multi-statement primer
5.2.4 Nested Loops

- Disjoint or fully nested
- Nested structures imply nested scopes

```java
for (int i = 0; i < max; i++)
    for (int j = 0; j < max; j++)
        display i and j
```

5.2.5 While Loops

- `while (condition)`
- `for (; condition;)
  
  body

5.3.1 Input Validation

Three ways for handling bad input:

- Crash
  - Primitive (but better than no validation)
- Print a message then end
  - Better. Requires an `else` statement to skip the rest of the program
- Print a message and allow retries
  - Best. Requires an `if` statement inside a loop
5.3.1 Input Validation

Three ways for handling bad input:

- **Crash**
  - Primitive (but better than no validation)

- **Print a message then end**
  - Better. Requires an `else` statement to skip the rest of the program

- **Print a message and allow retries**
  - Best. Requires an `if` statement inside a loop

---

5.3.2 File I/O

Key points to remember:

- Use `JFileChooser` to read file names
- File input is done through `Scanner`
- File output is done through `PrintStream`
- Always `close` the file after using it
- Suffix the main method with `throws java.io.IOException`
- Handle end-of-file by using:
  ```java
  for (; fileInput.hasNext();)
  ```