

---

---

---

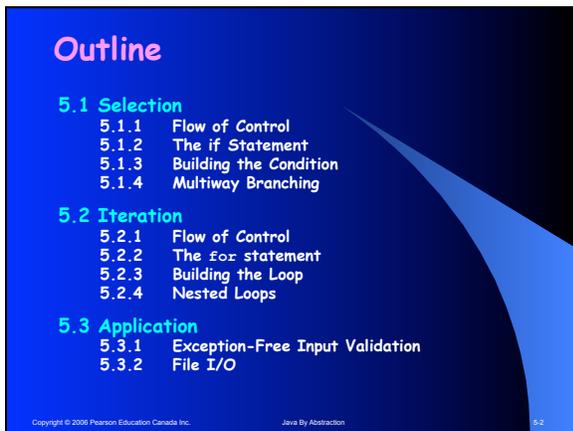
---

---

---

---

---



---

---

---

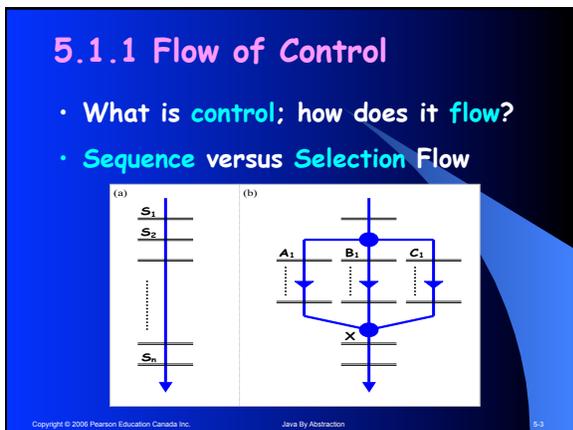
---

---

---

---

---



---

---

---

---

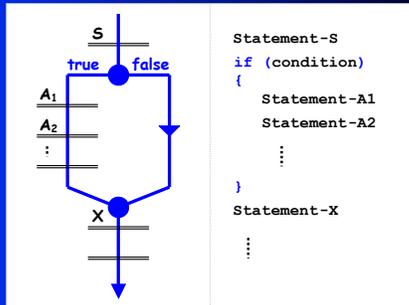
---

---

---

---

### 5.1.2 The if Statement



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

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5-4

---

---

---

---

---

---

---

---

### Example

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

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5-5

---

---

---

---

---

---

---

---

### Example

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

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

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5-6

---

---

---

---

---

---

---

---

### 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);
```

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.7

---

---

---

---

---

---

---

---

### The if-else Statement

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

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.8

---

---

---

---

---

---

---

---

### Example

Rewrite the abs value fragment using if-else

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.9

---

---

---

---

---

---

---

---

**Example**

Rewrite the abs value fragment using if-else

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

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.10

---

---

---

---

---

---

---

---

**Pitfall**

What is wrong with this?

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

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.11

---

---

---

---

---

---

---

---

**5.1.3 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 !

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.12

---

---

---

---

---

---

---

---

### Operator Precedence



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

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5-13

---

---

---

---

---

---

---

---

### Examples

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

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5-14

---

---

---

---

---

---

---

---

### Examples

- Express the condition  $x \in [a, b)$   
`if (x >= 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 >= b)`

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5-15

---

---

---

---

---

---

---

---

**Example**

Write a fragment to determine if the objects of two given references *x* and *y* are equal.

```
if (x.equals(y))
```

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.16

---

---

---

---

---

---

---

---

**Example**

Write a fragment to determine if the objects of two given references *x* and *y* are equal.

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

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

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.17

---

---

---

---

---

---

---

---

**5.1.4 Multiway Branching**

Reduce an *n*-way branch to *n*-1 2-way branches

```
Statement-S
if (condition-1)
{ Statement-A1
  Statement-A2
  ...
} else if (condition-2)
{ Statement-B1
  Statement-B2
  ...
} else
{ Statement-C1
  Statement-C2
  ...
}
Statement-X
```

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.18

---

---

---

---

---

---

---

---

### Example

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

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.19

---

---

---

---

---

---

---

---

### Example

```
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';  
}
```

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.20

---

---

---

---

---

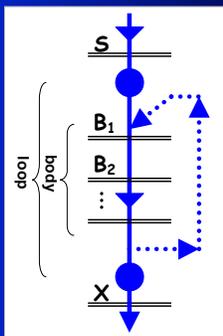
---

---

---

### 5.2.1 Flow of Control

I  
t  
e  
r  
a  
t  
i  
o  
n



Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.21

---

---

---

---

---

---

---

---

### 5.2.2 The for statement

**Flow:**

**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

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.22

---

---

---

---

---

---

---

---

### Example

```
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("\t"); // tab  
    output.println(sqrt);  
}
```

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.23

---

---

---

---

---

---

---

---

### for (initial; condition; bottom)

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

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

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.24

---

---

---

---

---

---

---

---

**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?

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.25

---

---

---

---

---

---

---

---

**for (initial; condition; bottom)**

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

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.26

---

---

---

---

---

---

---

---

**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
```

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.27

---

---

---

---

---

---

---

---

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

```
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.

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.28

---

---

---

---

---

---

---

---

**Example**  
Rewrite the fragment so that 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++)
{
}
output.println(expo - 1);
```

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

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.29

---

---

---

---

---

---

---

---

**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, ...

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.30

---

---

---

---

---

---

---

---

### Sentinel-Based Looping

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

Pseudo-code:

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

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.31

---

---

---

---

---

---

---

---

### Sentinel-Based Looping

Write a prog 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

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.32

---

---

---

---

---

---

---

---

### Sentinel-Based Looping

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

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

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.33

---

---

---

---

---

---

---

---

### Sentinel-Based Looping

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

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

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.34

---

---

---

---

---

---

---

---

### Sentinel-Based Looping

```
for (int n=input.nextInt(); not sentinel; n=input.nextInt())
{
    process the int
}
```

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

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.35

---

---

---

---

---

---

---

---

### Number Statistics

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

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.36

---

---

---

---

---

---

---

---

### 5.2.4 Nested Loops

- Disjoint or fully nested
- Nested structures imply nested scopes

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

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.37

---

---

---

---

---

---

---

---

### 5.2.5 While Loops

```
while (condition)  
{  
    body  
}
```

is the same as

```
for (; condition;)  
{  
    body  
}
```

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.38

---

---

---

---

---

---

---

---

### 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

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5.39

---

---

---

---

---

---

---

---

### 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

**Exception Free**

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5-40

---

---

---

---

---

---

---

---

### 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:  
`for (; fileInput.hasNextP();)`

Copyright © 2006 Pearson Education Canada Inc. Java By Abstraction 5-41

---

---

---

---

---

---

---

---