

# Week 3 - Decisions

**Stepwise Refinement**

**if - else**

**Conditional Operator**

**Switch statement**

3/13/01

1

# Initial Specification

**Write a program to manage withdrawals from a bank account. If there are enough funds, the program should accept the withdrawal, otherwise it should reject it.**

- **What is the initial value of the balance?**
- **What kind of outputs should the program produce?**

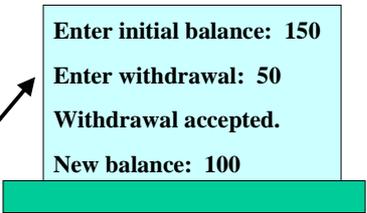


3/13/01

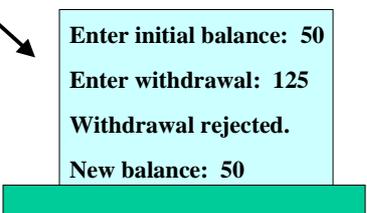
2

## Refining the Specifications

- The program should accept the initial balance, and then accept a withdrawal.
- If there are enough funds, the program should accept the withdrawal, otherwise it should reject it.
- The program should display the new balance at the end of the transaction.
- Input comes from the keyboard and output goes to the screen.



Enter initial balance: 150  
Enter withdrawal: 50  
Withdrawal accepted.  
New balance: 100



Enter initial balance: 50  
Enter withdrawal: 125  
Withdrawal rejected.  
New balance: 50

3/13/01

3

## Initial Design & Refinement

### Get balance

- display prompt
- get balance value

### Get withdrawal

- display prompt
- get withdrawal value

### Decide accept or refuse

- if balance greater than or equal to withdrawal, accept
  - display accept message
  - calculate new balance
  - display new balance message and value
- otherwise reject
  - display reject message
  - display new balance message and value

3/13/01

4

# Implementation

```
import java.io.*;
class AccountManagement {
    public static void main (String[] args) throws IOException {
        BufferedReader stdin = new BufferedReader
            (new InputStreamReader (System.in));
        String initString, withdrawString;
        int initialBalance, withdrawal;

        System.out.println ("Enter initial balance");
        initString = stdin.readLine();
        initialBalance = Integer.parseInt (initString);
        System.out.println ("Enter withdrawal");
        withdrawString = stdin.readLine();
        withdrawal = Integer.parseInt (withdrawString);
```

3/13/01

5

# Implementation

```
if (initialBalance >= withdrawal) {
    System.out.println("Withdrawal accepted");
    System.out.print("New balance: ");
    System.out.println(initialBalance - withdrawal);
}
else {
    System.out.println("Withdrawal rejected");
    System.out.print("New balance: ");
    System.out.println(initialBalance);
}
} // end of main()
} // end of class
```

3/13/01

6

## In the last program we used a ...

- the **if ... else** statement
- the “greater than or equal to” operator **>=**
- the use of curly braces **{** and **}** to make a *block*
  - *so several statements are considered as one.*
  - *could be empty as in { }*

```
if (initialBalance >= withdrawal) {
    System.out.println("Withdrawal accepted");
    System.out.print("New balance: ");
    System.out.println(initialBalance - withdrawal);
}
```

3/13/01

7

## The **if...else** statement

- main statement for decision making
- Its general form is:

```
if (condition)
    statement // executed if condition true
else statement // executed if condition false
```

- or, if using blocks,

```
if (condition) {
    statements; // executed if condition true
}
else {
    statements; // executed if condition false
}
```

3/13/01

8

## The rules ...

- **condition** must be a **boolean** condition - an expression that evaluates to **true** or **false**.
- If **true** the first statement is executed; otherwise the second statement is executed.
- Statements can be more than one if they are included in a block.
- The individual **statements** to be executed are statements, so they include a **;** at the end;
- the **else** is optional. If not present, the syntax is:

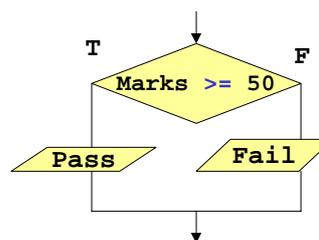
```
if (condition) {  
    statements;  
}
```

3/13/01

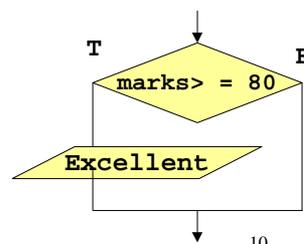
9

## Comparing if and if ... else

```
// prints Pass if marks >= 50  
// prints Fail otherwise  
if (marks >= 50)  
    System.out.println("Pass");  
else  
    System.out.println("Fail");
```



```
// prints Excellent if marks >= 80  
if (marks >= 80)  
    System.out.println("Excellent");
```



3/13/01

10

# Quiz

The following program segment is attempting to print messages based on CS108 final marks. Identify all the errors

```
if marks >= 50;  
    System.out.println("You passed CS 108");  
    System.out.println("You may proceed to CS109");  
else  
    System.out.println("You failed CS108");  
    System.out.println("You have to repeat CS108");
```

3/13/01

11

# Quiz

```
Enter the amount in cents:  
-357
```

What if the user enter a -ve number amount ?  
(you are not required to handle!)

```
if ( amount < 0){  
    System.out.println(.....);  
    System.exit(1);  
}
```

3/13/01

12

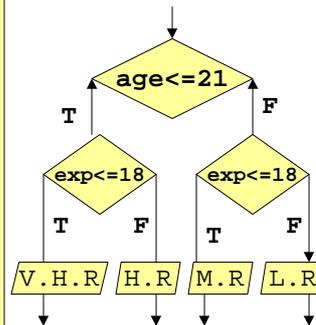
## Nested if ... else

Imagine that you have joined a motor insurance company asked to print a chart based on age and experience (months)

	exp <= 18 m	exp > 18 m
age <= 21	Very High Risk	High Risk
age > 21	Moderate Risk	Low Risk

```

if (age <= 21)
    if (exp <= 18)
        System.out.println("V.H.R")
    else
        System.out.println("H.R")
else
    if (exp <= 18)
        System.out.println("M. Risk");
    else
        System.out.println("L. Risk");
    
```



13

## Nested if ... else

```

if (mark >= 50)    // pass the subject
    if (mark >= 80)
        System.out.println("Well done!");
    else
        System.out.println("Passed the subject");
    
```

```

mark = 85    Well done!
mark = 65    Passed the subject
mark = 25    ____ (no output)
mark = 125   Well done!
    
```

Note that else is associated with the closest if

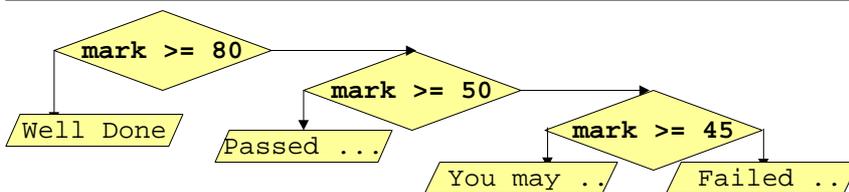
How can we modify the program to print Failed the subject if mark < 50 ?

3/13/01

14

## Another way

```
if (Mark >= 80)
    System.out.println("Well done!");
else if (Mark >= 50)
    System.out.println("Passed the subject");
else if (Mark >= 45)
    System.out.println("You may get another chance!");
else
    System.out.println("Failed the subject");
```



Quiz What would you change in the above program to print Incorrect Marks if mark > 100 ?

3/13/01

15

## Summary of Decision Making

```
if (Boolean_expression)
    statement;           // simple decision
```

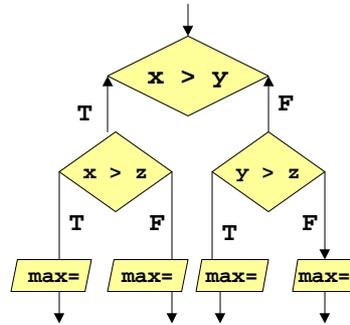
```
if (Boolean_expression)
    statement;
else
    statement;           // else is optional
```

```
if (Boolean_expression1)
    statement;
else if (Boolean_expression2)
    statement;           // 0 or more else if
else
    statement;           // else is optional
```

16

## Quiz - Nested If

```
if( x > y )
  if ( x > z )
    max = _; // 1
  else
    max = _; // 2
else
  if ( y > z )
    max = _; // 3
  else
    max = _; // 4
```



Fill in the 4 blanks labelled 1 to 4.

## Conditional Operator

The conditional operator is used as a shorthand for an `if...else` statement.

```
if (x < y)
  minVal = x;
else
  minVal = y;
```

Equivalent statement using Conditional operator

```
minVal = (x < y) ? x : y;
```

The general case is:

```
condition ? yesExpression : noExpression;
```

The condition is evaluated first; if `true` the value of the entire expression is `yesExpression`, otherwise the value is `noExpression`.

## What is the output ?

```
System.out.println("The number is: "  
    + ((n % 2) == 0 ? "even" : "odd"));
```

3/13/01

19

## The **switch** statement

Java **switch** statement for multiway decisions, is an alternative to nested **if...else**.

The following code prints the month name corresponding to a month number:

```
switch (month) {  
    case 1 : System.out.println("January");  
        break;  
    case 2 : System.out.println("February");  
        break;  
    case 3 : System.out.println("March");  
        break;  
    case 4 : System.out.println("April");  
        break;  
    case 5 : System.out.println("May");  
        break;
```

3/13/01

20

```
case 6 : System.out.println("June");
    break;
case 7 : System.out.println("July");
    break;
case 8 : System.out.println("August");
    break;
case 9 : System.out.println("September");
    break;
case 10 : System.out.println("October");
    break;
case 11 : System.out.println("November");
    break;
case 12 : System.out.println("December");
    break;
default : System.out.println("Not a valid month");
    break;
}
```

### output

```
Enter a month(1-12):
5
May
```

3/13/01

21

## Rules for `switch` statement

- The selector must be an integer-type expression
- If the selector matches any of the values in the cases, the execution continues there.
- The **break** is needed to avoid the execution of the statements following the match. After the break execution continues with the first statement after the switch.
- If there is no match on any of the cases, control transfers to a **default** case.
- If there is no **default** case the entire **switch** is terminated and execution continues with the first statement after the **switch**.

3/13/01

22

## Handling several alternatives with Switch

```
switch (month) {
  case 1 :
  case 2 :
  case 3 : System.out.println("First Quarter");
            break;
  case 4 :
  case 5 :
  case 6 : System.out.println("Second Quarter");
            break;
  case 7 :
  case 8 :
  case 9 : System.out.println("Third Quarter");
            break;
  case 10 :
  case 11 :
  case 12 : System.out.println("Fourth Quarter");
             break;
  default : System.out.println("Wrong Input!");
            break;
}
```

months 1,2 3  
fall through to  
this  
statement.

3/13/01

23

### Chap 2 Tutes Q7.

Let **b** have the value 5 and **c** have the value 8.  
What are the values of **a**, **b** and **c** after each line  
of the following fragment

```
a = b++ + c++;
a = b++ + ++c;
a = ++b + c++;
a = ++b + ++c;
```

3/13/01

24

### Chap 2 Tutes Q8

The following program segment printed an incorrect output (instead of the expected *sqr = 10000000000*). Explain why, and state how the program segment may be corrected.

```
int num1 = 100000;  
int sqr = num1 * num1;  
System.out.println("sqr = "+sqr);
```

3/13/01

25

### Chap 2 Tutes Q9

Given the declarations x,y and z are int, float and double variables which of the following statement(s) are likely to result in error ?

- (a)  $x = y$ ;      (b)  $x = z$ ;      (c)  $y = x$ ;  
(d)  $y = z$ ;      (e)  $z = x$ ;      (f)  $z = y$ ;

3/13/01

26

### Chap 2 Tutes Q11.

The statements below are equivalent to:

```
y = x;  
x = x+1;
```

- (a) `y = x++;`
- (b) `x = y++;`
- (c) `y = ++x;`
- (d) `x = ++y;`

3/13/01

27

### Chap 2 Tutes Q12.

Trace through the program segment below and state the output if x and y were originally set to 5 and 8. Explain why it has failed to swap the values stored in x and y. Suggest how the code segment can be modified to swap them correctly.

```
int x = 5;  
int y = 8;  
// attempting to swap the values  
x = y;  
y = x;  
System.out.println("After swapping:x is "  
                    + x + " and y is " + y);
```

3/13/01

28