Understanding class definitions 3

Selection – if statements

- Produced by: Ms. Mairead Meagher, Ms. Siobhán Roche.

Ollscoil Teicneolaíochta an Oirdheiscirt
South East Technological University

Department of Computing and Mathematics
http://www.setu.ie/

Upcoming

- Conditional statements;
  - When do we need them
  - Syntax
  - If stmt, if-else stmts
  - Examples
- Local variables. (next slidedeck)

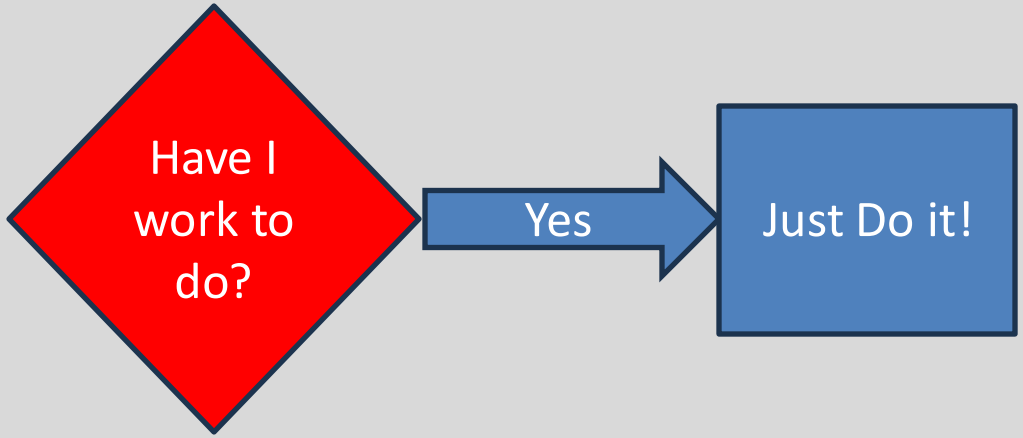
Reflecting on the ticket machines

- Their behavior is inadequate in several ways:
  - No checks on the amounts entered.
  - No refunds.
  - No checks for a sensible initialization.
- How can we do better?
  - We need the ability to choose between different courses of action.

Conditionals – the if statement

Making choices in everyday life

- If I have an assignment to complete, then I shall work on my assignment



Conditional Statement Syntax (1)

```
if(condition1…perform some test)
{
     Do these statements if the test gave a true result
}
```

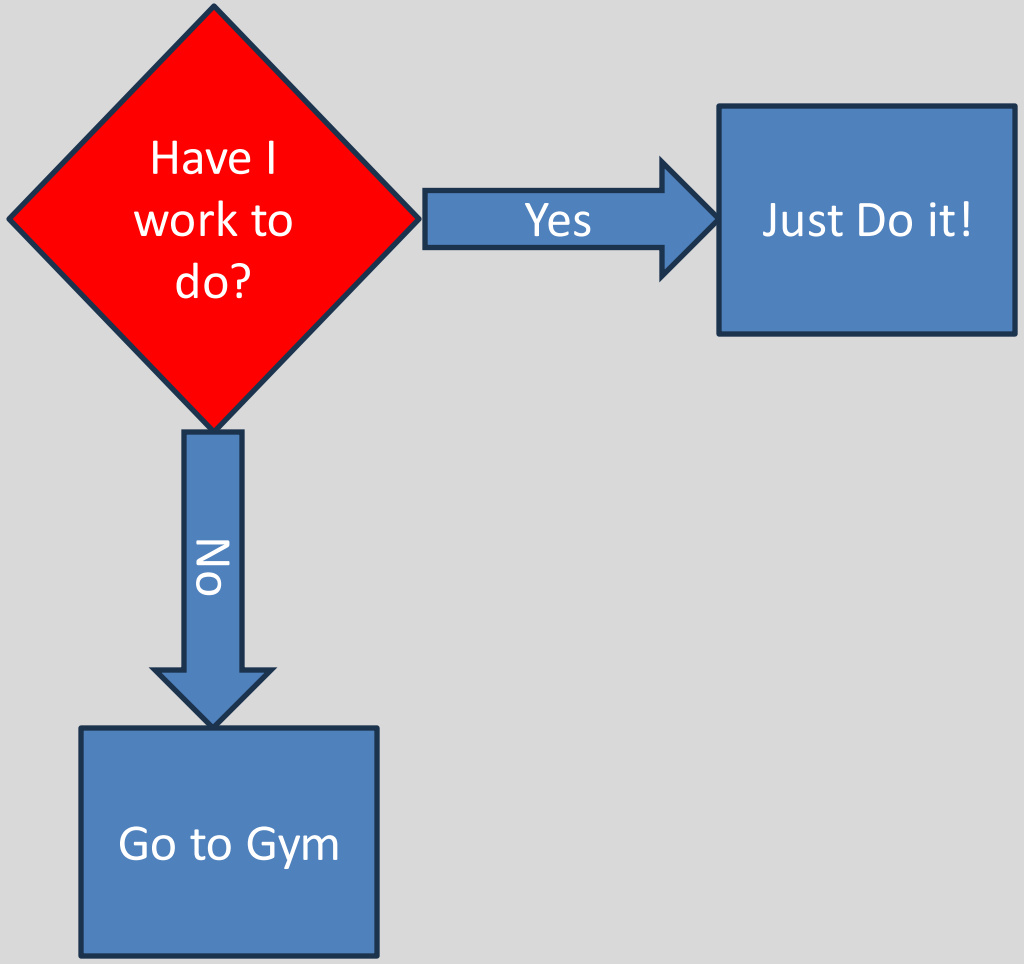
Making a choice in the ticket machine (1)

```
public void insertMoney(int amount)
{
    if(amount > 0) {
        balance = balance + amount;
    }
}
```

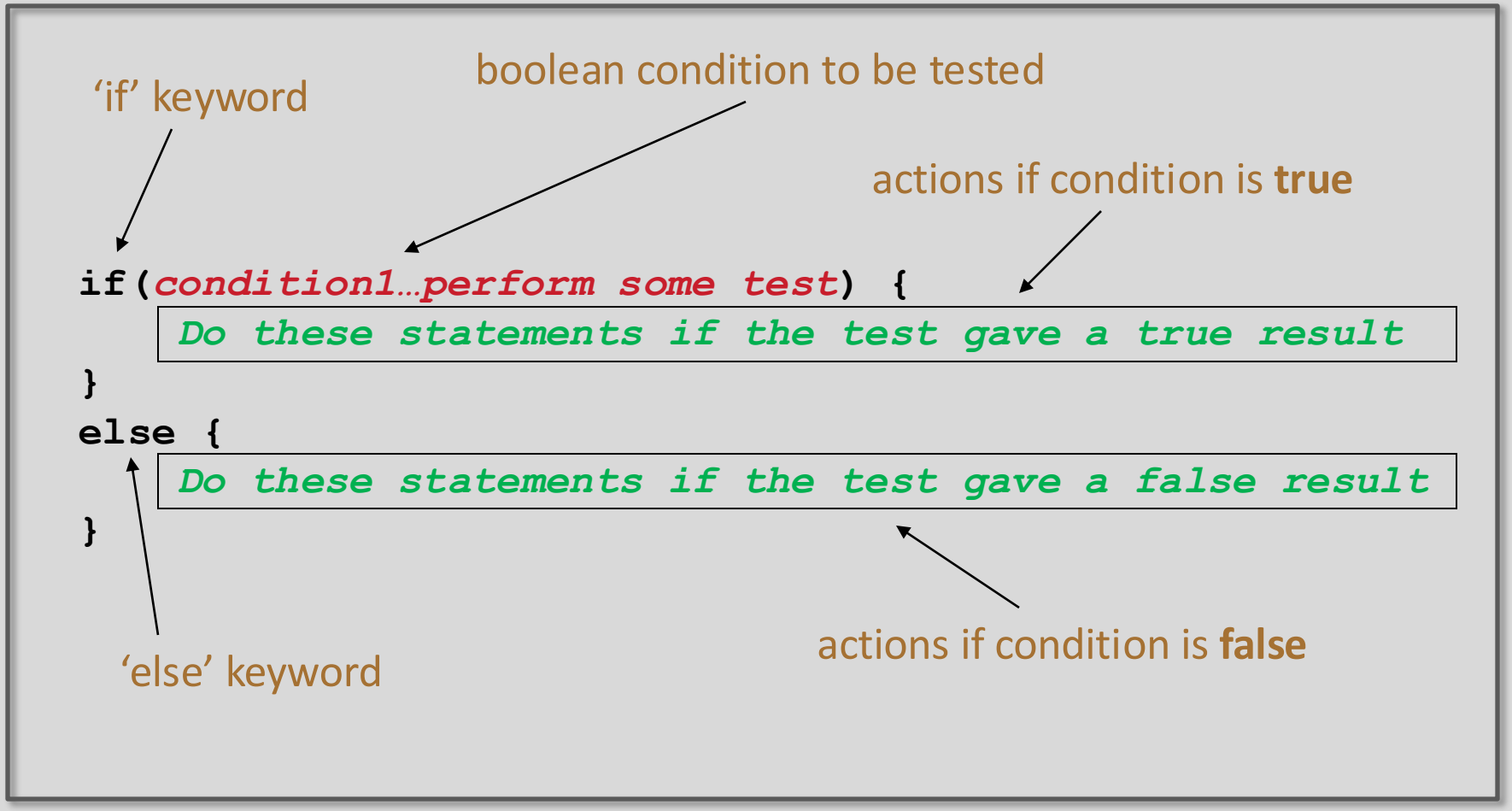
the conditional statement avoids an inappropriate action

Making choices in everyday life (2)

- If I have an assignment to complete, then I shall work on my assignment
- Otherwise I will go to the Gym



Conditional Statement Syntax (2)



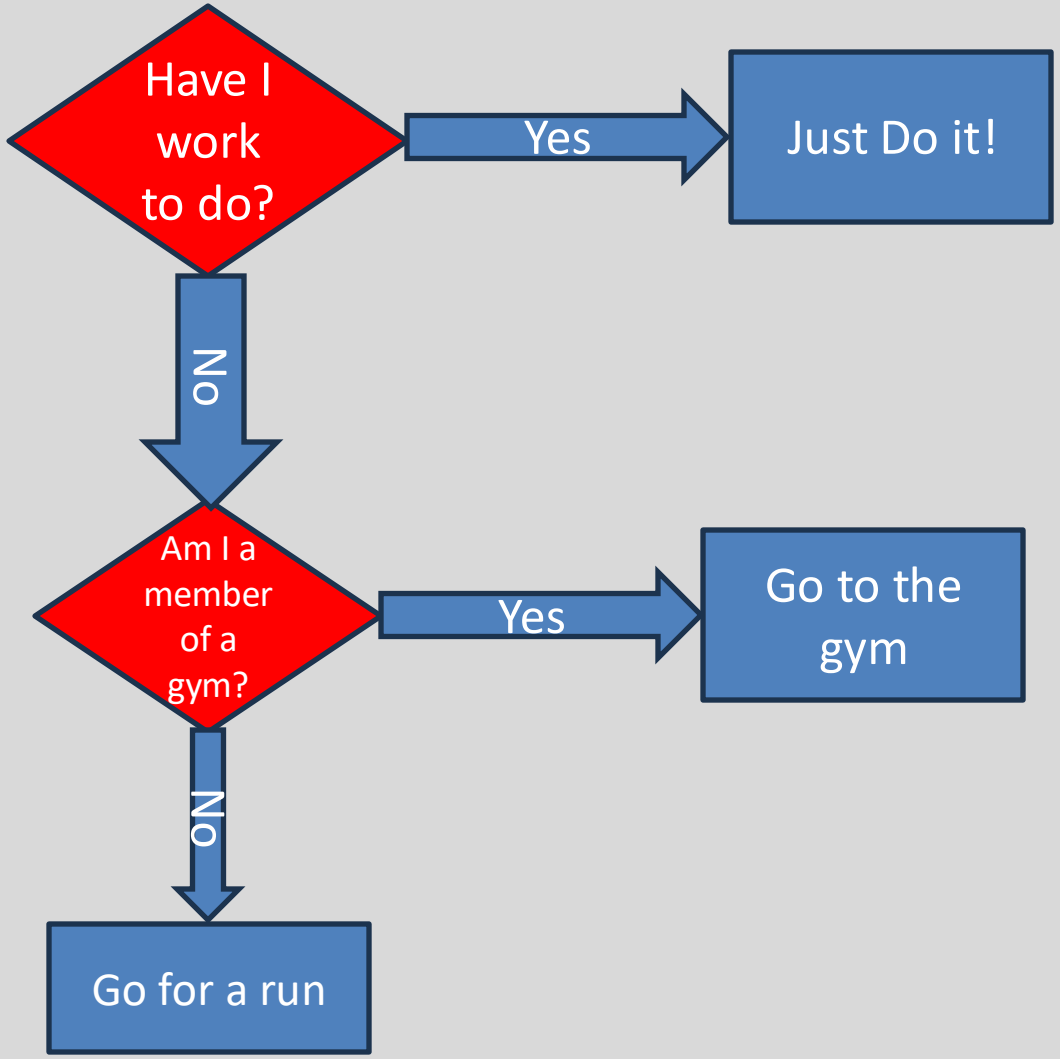
Making a choice in the ticket machine (2)

```
public void insertMoney(int amount)
{
    if(amount > 0) {
        balance = balance + amount;
    }
    else {
        System.out.printf(
           "Use a positive amount: %d%n",
            amount);
    }
}
```

the conditional statement avoids an inappropriate action

Making choices in everyday life (3)

- If I have an assignment to complete, then I shall work on my assignment
- Otherwise if I am a member of a gym, go to the gym
- Otherwise I will go for a run



Conditional Statement Syntax (3)

```
if(condition1…perform some test)
{
    Do these statements if condition1 gave a true result
}
else if(condition2…perform some test)
{
    Do these statements if condition1 gave a false
    result and condition2 gave a true result
}
else
{
    Do these statements if both condition1 and
    condition2 gave a false result
}
```

Making a choice in the ticket machine (3)

```
public void specialOffer(int amount)
{
    if(amount >100) {
        balance = balance + 50;
    }
    else if amount > 50{
        balance = balance + 25;
    }
    else amount = amount + 5;
}
```

Note that if any condition is true, the associated statements are executed AND the if statement is finished.

Boolean conditions

- A boolean condition is an expression that evaluates to either **true** or **false** e.g.

  price < 50

- An if statement evaluates a **boolean condition** and its result will determine which portion of the if statement is executed.

Boolean conditions

```
// Do these statements before.

if (boolean condition)
{
    // Perform this clause if the
        // condition is true.
}

// Do these statements after.
```

Java Relational Operators

Operator	Use	Returns true if…
>	op1 > op2	op1 is **greater** than op2
>=	op1 >= op2	op1 is **greater than or equal** to op2
<	op1 < op2	op1 is **less** than to op2
<=	op1 <= op2	op1 is **less than or equal** to op2
==	op1 == op2	op1 and op2 are **equal**
!=	op1 != op2	op1 and op2 are **not equal**

BEWARE = is an assignment operator.
It doesn't test for equality. Use == to test for equality in primitive types

Source: http://www.freejavaguide.com/relational_operators.htm

Some notes on the if statement

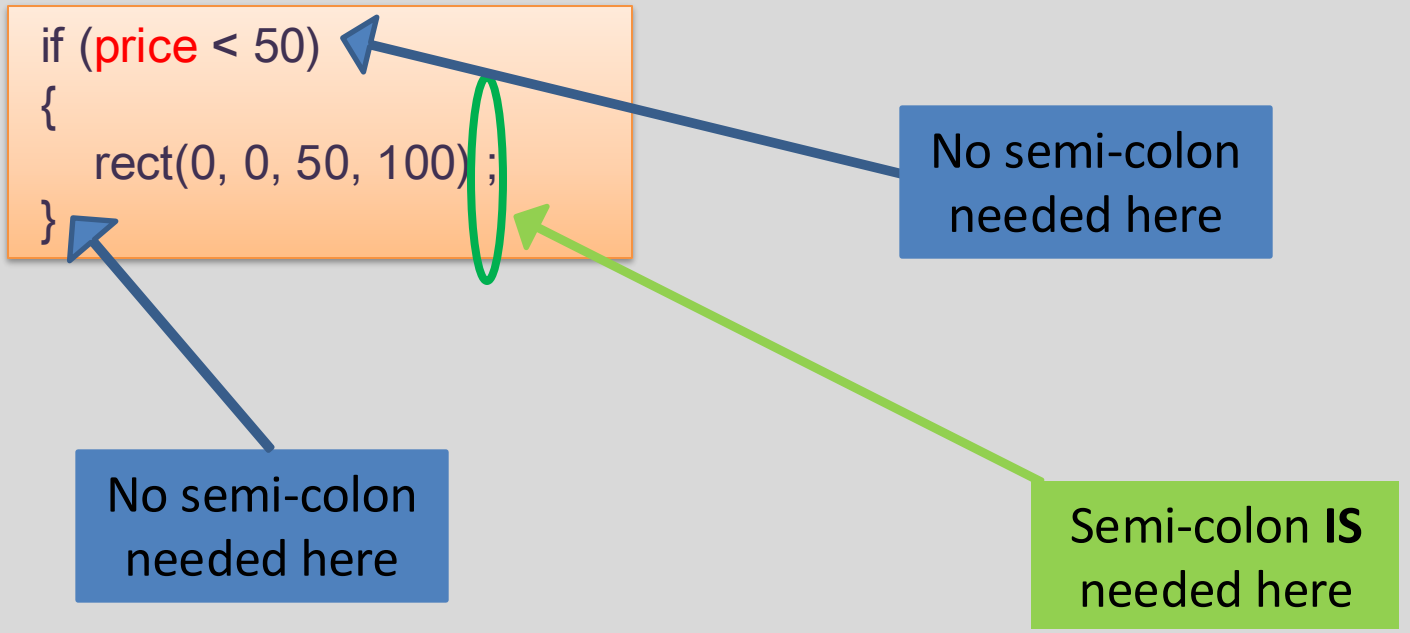
- An if statement IS a **statement**;
  it is only executed once.
- When your if statement only has one statement inside it, you do not need to use the curly braces.
- For example, both of these are the same:

```
if (price < 50)
{
   price = 50;
}
```

```
if (price < 50)
   price = 50;
```

Some notes on the if statement

- The semi-colon (;) is a **statement terminator**.



Questions?