



ARITHMETIC AND LOGICAL OPERATORS, SELECTION

Recap:

<i>Type</i>	<i>Variable</i>	<i>Value</i>
double	volume	= 2.4;



Truncation and promotion

double x, y = 7.0, z = 2.0;

x = y / z;

3.5 3.5

int x, y = 7, z = 2;

x = y / z;

3 3

double x;

int y = 7, z = 2;

x = y / z;

3.0 3

double x, y = 7.0;

int z = 2;

x = y / z;

3.5 3.5

*double has a higher priority than an
int, so the int is promoted to a double*



Casting a type

```
double x, z = 2.0;
```

```
int y = 7;
```

```
x = y / (int)z;
```

3.0 **3**

*double cast
to an int*

The 'modulus' operator

MO: returns the **remainder**
of integer division

```
int x, y = 7, z = 2;
```

```
x = y / z;    // x = 3
```

```
x = y % z;    // x = 1
```



The primitive type **char**

Type **char**: a character that *also can be treated as a integer*

The ASCII character set defines numbers (codes) that correspond to electronic characters

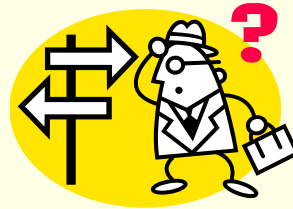
```
Console.WriteLine("Letter is " + (int)let );
```

Letter is 97

0..	
97	a
98	b
99	c
..127	



SELECTION – MAKING A DECISION



Program flow of execution governed by
CONTROL STRUCTURES

Selection: if-else control structures

Iteration: loop structures

Control structures use **relational operators**



Relational operators evaluate as **true** or **false**

<i>condition</i>	<i>meaning</i>
<code>a == b</code>	a equals b
<code>a != b</code>	a not equal to b
<code>a > b</code>	a greater than b
<code>a <= b</code>	a less than or equal to b
<code>&&</code> <code> </code> <code>!</code>	and, or, not

Multiple conditions can be tested:

e.g. `(a==b) || (a != c)`

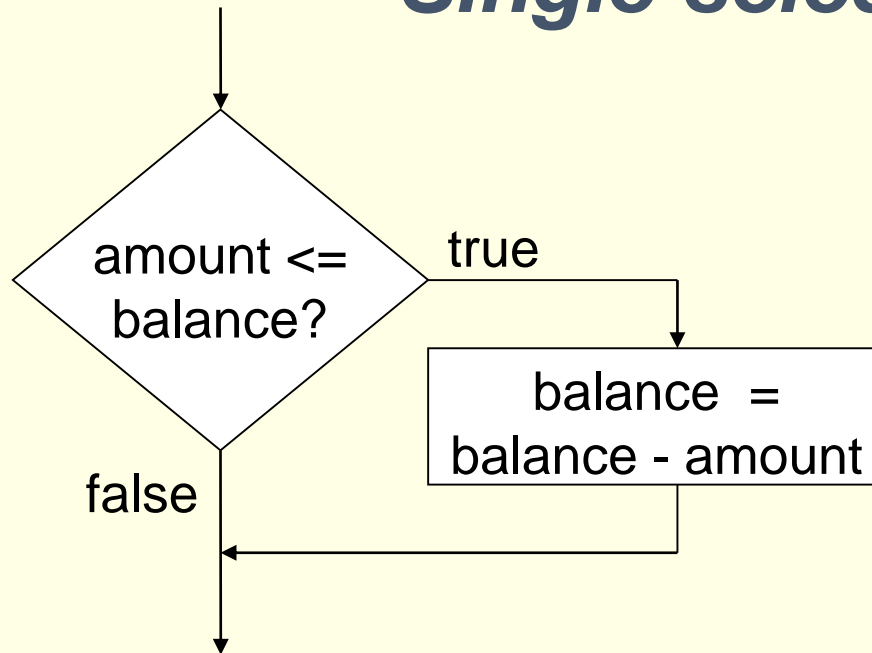
Note 1: `a = b;` is quite different from `a == b;`

Note 2: be careful with equality tests on type double



SELECTION: if, if-else, switch

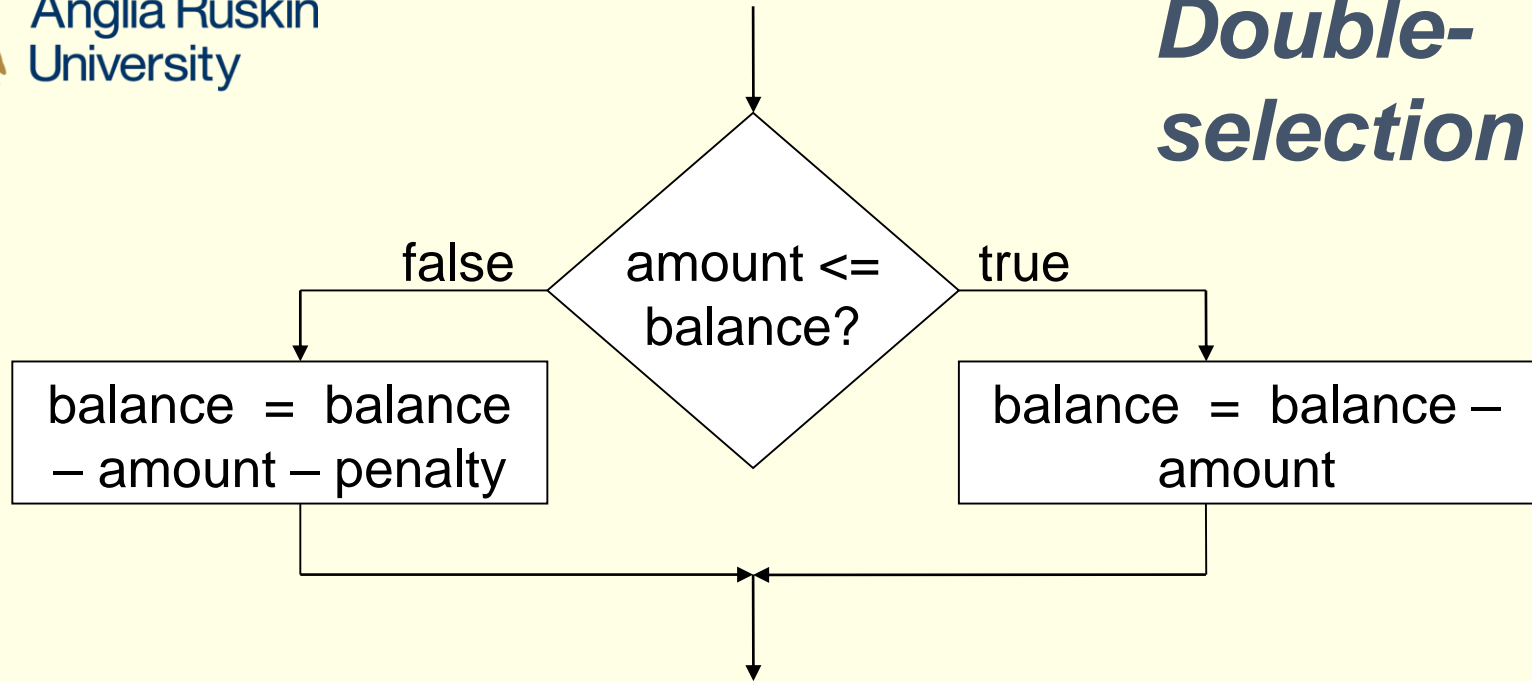
Single-selection



```
if ( amount <= balance )  
{  
    balance = balance - amount;  
}
```



Double-selection



```
if ( amount <= balance )
{
    balance = balance - amount;
}
else
{
    balance = balance - amount - penalty;
}
```




Multiple- if/else selection

```
if ( amount <= balance )  
{  
    balance = balance - amount;  
}  
else if ( balance < -300 );  
{  
    System.out.println("No funds available");  
}  
else  
{  
    balance = balance - amount - penalty;  
}
```



Multiple-selection for equality using SWITCH

```
switch( value )  
{  
    case a:           // value == a  
        statements;  
        break;  
    case b:           // value == b  
        statements;  
        break;  
    case c:           // value == c  
        statements;  
        break;  
    default:         // value != a, b or c  
        statements;  
}
```



```
leapyear = year % 4;  
switch( leapyear )  
{  
    case 0:  
        Console.WriteLine( year + "is a leap year");  
        break;  
    case 1:  
        Console.WriteLine("3 years to go");  
        break;  
    case 2: case 3:  
        Console.WriteLine("1 or 2 years to go");  
        break;  
    default:  
        Console.WriteLine("A strange year indeed");  
}
```



Conclusions

- int/int division will **truncate** a fraction
- The type of a variable can temporarily change either through ***promotion*** or ***casting***
- Operators include> **arithmetical**, **relational** and **logical** kinds
- A control structure controls the **flow of execution** in a program
- Selection control can be **if**, **if-else**, or **switch**