CS620c Structured Programming Lesson 7

Joe Duffin

Email: Joseph.Duffin@nuim.ie

More on Numeric Operators

Operator	**Result**
+	Addition
-	Subtraction
*****	Multiplication
/	Division
+=	Addition assignment
-=	Subtraction assignment
*=	Multiplication assignment
/=	Division assignment
%	Modulus We'll look at this again
++	Increment NEW
--	Decrement NEW

Any questions?

Increment and Decrement

- Increment (++) adds one to the variable and stores the new value in that variable
  - The following two lines achieve the same effect:

```
num = num+1;
num++;
```

- Decrement (--) subtracts one from the variable and stores the new number in that variable
  - The following two lines achieve the same effect:

```
num = num-1;
num--;
```

Modulus Revisited

- %   Modulus (remainder)
- Returns the remainder after division.
- Can be used on real and whole numbers
- Remember:

```
int quotient = 7 / 3; // yields 2
int remainder = 7 % 3; // yields 1
```

- You can use modulus to check whether one number is divisible by another: if x % y is zero, then x is divisible by y.
- Also, you can use the modulus operator to extract the rightmost digit or digits from a number. For example, x % 10 yields the rightmost digit of x.

Breaking up a number

- Let’s revisit the notion of breaking up a number using modulus and division..
- Think about an odometer in a car.. (an odometer measures the distance travelled)

0	2	3

- 23 = 2*10 +3
- There are two lots of 10 in 20
- There are no lots of 10 in 3
- If I divide 23/10 I get two with a remainder of 3

Breaking up a number (2)

- What is the number 47 composed of?
  - 4 *10 + 7 *1

- What is the number 123 composed of?
  - 1 *100 + 2*10 + 3*1 =
  - 1*(10*10)+ 2*(10)+ 3*1
  - Thus there are 12 lots of ten in 123 and the remainder is 3

- What is the number 1234 composed of?
  - 1*1000 + 2*100+3*10+4
  - 1*(10*10*10)+2*(10*10)+3*(10)+4
  - Thus there are 123 lots of ten in 1234 and the reminder is 4

Example

- So to break up any number you need to use modulus and division and the magic number 10!
- Let’s do an example
- Write an algorithm that breaks up a three-digit number into it’s component parts e.g. if the number is 236, the program should print the following:

```
The first digit is 2
The second digit is 3
The third digit is 6
```

Problem suggestion.

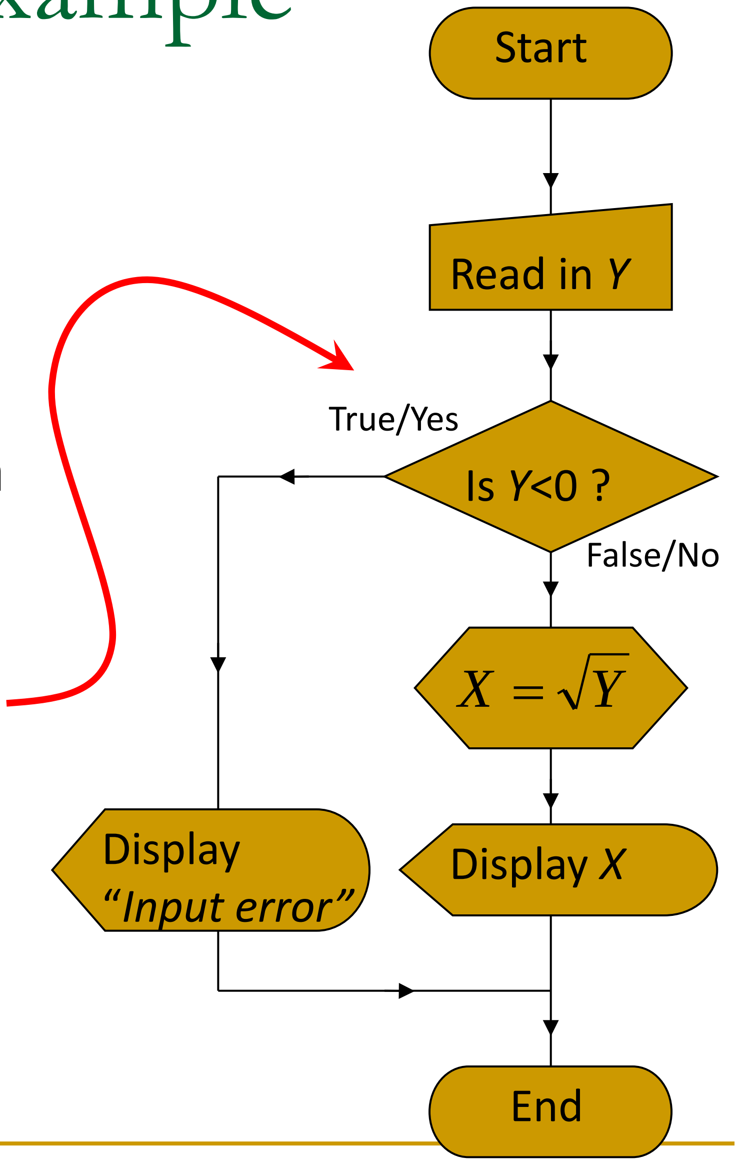
- Write an algorithm and a program to break up the digits of a 5-digit number into its component parts. (use whatever variables you think that you may need).

Selection Statements

- All but the most trivial computer programs need to make decisions.
- They test a condition and operate differently based on the outcome of the test.
- This is quite common in real life…
- All programming languages have some form of an if statement that tests conditions.

Selection Statement Example

- Sometimes we would like to choose the code to run based on some decision made during the programs execution.
- A <u>condition</u> is used to make a decision as to which direction the program will <u>branch</u> during execution.
- The flow chart shows how a condition can be used to prevent user input causing a crash when a negative value is entered.



Types of Selection Statements

- if (<condition>) { <true, statement(s)>; }

- if (<condition>) { <true, statement(s)>; } else { <false, statement(s)>; }

- if (<condition1>) { <true condition1, statement(s)>; }
  else if (<condition2>) { < true condition2 and not condition 1, statement(s);> }
  else { < not condition2 and not condition 1, statement(s)> }

- switch(<integer variable>)
  {
    case <num1> : <case num1, statements>; break;
    case <num2> : <case num2, statements>; break;
    default: <non of the cases, statements>;

}

- num = (<condition>) ? value_if_true : value_if_false;

Simple if statement

```
if(condition){
    statement;
    :
 }
```

- If the condition is true everything between the opening and closing curly braces is executed

if else Statement

```
if(condition){
        statement1;
          :
}
else{
        statement2;
          :
}
```

- If the condition is true everything between the opening and closing curly braces is executed
- If the condition is false everything in the else block is executed (between the opening and closing curly braces after the word else)