COMP1917: 04 Loops

Sim Mautner

s.mautner@unsw.edu.au

August 2, 2016

References

- Moffat, Chapter 4: Loops
- Solutions to exercises in these slides:
  www.cse.unsw.edu.au/~simm/lectures/wk2-loops/

Motivation

- Ex 1: Write an application which prints out "Hello World" 5 times.

A Better Way...

1. Set a variable (i) to equal 0.
2. Repeat steps 3 and 4 while i is not equal to 5:
    1. Print "Hello World".
    2. Add 1 to i.

i	Output

A Better Way...

1. **Set a variable (i) to equal 0.**
2. Repeat steps 3 and 4 while i is not equal to 5:
    1. Print "Hello World".
    2. Add 1 to i.

i	**Output**
0	

A Better Way...

1. Set a variable (i) to equal 0.
2. **Repeat steps 3 and 4 while i is not equal to 5:**
    1. Print "Hello World".
    2. Add 1 to i.

i	**Output**
0	

A Better Way...

1. Set a variable (i) to equal 0.
2. Repeat steps 3 and 4 while i is not equal to 5:
    1. **Print "Hello World".**
    2. Add 1 to i.

i	**Output**
0	Hello World

A Better Way...

1. Set a variable (i) to equal 0.
2. Repeat steps 3 and 4 while i is not equal to 5:
    1. Print "Hello World".
    2. **Add 1 to i.**

i	**Output**
0	Hello World
1	

A Better Way...

1. Set a variable (i) to equal 0.
2. **Repeat steps 3 and 4 while i is not equal to 5:**
    1. Print "Hello World".
    2. Add 1 to i.

i	**Output**
0	Hello World
1	

A Better Way...

1. Set a variable (i) to equal 0.
2. Repeat steps 3 and 4 while i is not equal to 5:
    1. **Print "Hello World".**
    2. Add 1 to i.

i	**Output**
0	Hello World
1	Hello World

A Better Way...

1. Set a variable (i) to equal 0.
2. Repeat steps 3 and 4 while i is not equal to 5:
    1. Print "Hello World".
    2. **Add 1 to i.**

i	**Output**
0	Hello World
1	Hello World
2	

A Better Way...

1. Set a variable (i) to equal 0.
2. **Repeat steps 3 and 4 while i is not equal to 5:**
    1. Print "Hello World".
    2. Add 1 to i.

i	**Output**
0	Hello World
1	Hello World
2	

A Better Way...

1. Set a variable (i) to equal 0.
2. Repeat steps 3 and 4 while i is not equal to 5:
    1. **Print "Hello World".**
    2. Add 1 to i.

i	**Output**
0	Hello World
1	Hello World
2	Hello World

A Better Way...

1. Set a variable (i) to equal 0.
2. Repeat steps 3 and 4 while i is not equal to 5:
    1. Print "Hello World".
    2. **Add 1 to i.**

i	**Output**
0	Hello World
1	Hello World
2	Hello World
3	

A Better Way...

1. Set a variable (i) to equal 0.
2. **Repeat steps 3 and 4 while i is not equal to 5:**
    1. Print "Hello World".
    2. Add 1 to i.

i	**Output**
0	Hello World
1	Hello World
2	Hello World
3	

A Better Way...

1. Set a variable (i) to equal 0.
2. Repeat steps 3 and 4 while i is not equal to 5:
    1. **Print "Hello World".**
    2. Add 1 to i.

i	**Output**
0	Hello World
1	Hello World
2	Hello World
3	Hello World

A Better Way...

1. Set a variable (i) to equal 0.
2. Repeat steps 3 and 4 while i is not equal to 5:
    1. Print "Hello World".
    2. **Add 1 to i.**

i	**Output**
0	Hello World
1	Hello World
2	Hello World
3	Hello World
4	

A Better Way...

1. Set a variable (i) to equal 0.
2. **Repeat steps 3 and 4 while i is not equal to 5:**
    1. Print "Hello World".
    2. Add 1 to i.

i	**Output**
0	Hello World
1	Hello World
2	Hello World
3	Hello World
4	

A Better Way...

1. Set a variable (i) to equal 0.
2. Repeat steps 3 and 4 while i is not equal to 5:
    1. **Print "Hello World".**
    2. Add 1 to i.

i	**Output**
0	Hello World
1	Hello World
2	Hello World
3	Hello World
4	Hello World

A Better Way...

1. Set a variable (i) to equal 0.
2. Repeat steps 3 and 4 while i is not equal to 5:
    1. Print "Hello World".
    2. **Add 1 to i.**

i	**Output**
0	Hello World
1	Hello World
2	Hello World
3	Hello World
4	Hello World
5	

A Better Way...

1. Set a variable (i) to equal 0.
2. **Repeat steps 3 and 4 while i is not equal to 5:**
    1. Print "Hello World".
    2. Add 1 to i.

i	**Output**
0	Hello World
1	Hello World
2	Hello World
3	Hello World
4	Hello World
5	

Components of a Loop

There are 4 main components of every loop:

1. Initialisation: Where to start. Often a counter equalling a value, or an initial input.
2. Condition: How to know if it should repeat again, or stop.
3. Code to repeat: The code that we want to do multiple times.
4. Update: Change a value so that the loop eventually terminates.

```
int i=0; // <-- Initialisation
while( /* condition */ ) { // <-- Condition

   // Code goes here     <-- Code to repeat

   i=i+;  // <-- Update
}
```

Loops Exercises

- Ex 2: Write an application which prints out "Hello World" 5 times.
- Ex 3: Write an application which asks the user to enter a number, and prints out "Hello World" that number of times.
- Ex 4: Write an application which asks the user to enter a number between 1 and 100 inclusive. If the user enters a number outside of that range, display an error message and ask again until the user enters a valid number.

Shorthand

- Instead of this:

  ```
  i = i+1;
  ```

  we can use this:

  ```
  i++;
  ```

- Instead of this:

  ```
  i = i-1;
  ```

  we can use this:

  ```
  i--;
  ```

- They do the same thing. Either one is fine to use.

Loops Exercises

- Ex 5: Write an application which asks the user for a number, and then prints out every number from 0 up to that number.
- Ex 6: Write an application which asks the user for a number, and then prints out every number from that one down to 0.

Nested Loops

- Like if-statements, loops can go within each other.
- Loops can also go inside if-statements and vice versa.

Nested Loops Exercises

- Ex 7: Write an application which asks the user for a number, and then prints out a square of x's of that dimension.