
Algorithm Discovery

Topics:

Iterative Operations
Algorithmic Problem Solving

Prof. Dr. Slim Abdennadher

15.11.2007

What is life?

“Life is just one damn thing after another.”

Mark Twain

“Life isn’t just one damn thing after another ... it is the same damn thing over and over again.”

Edna St. Vincent Millay

Iterative Operation – While

Repeat a set of steps over and over – also called a **looping operation**

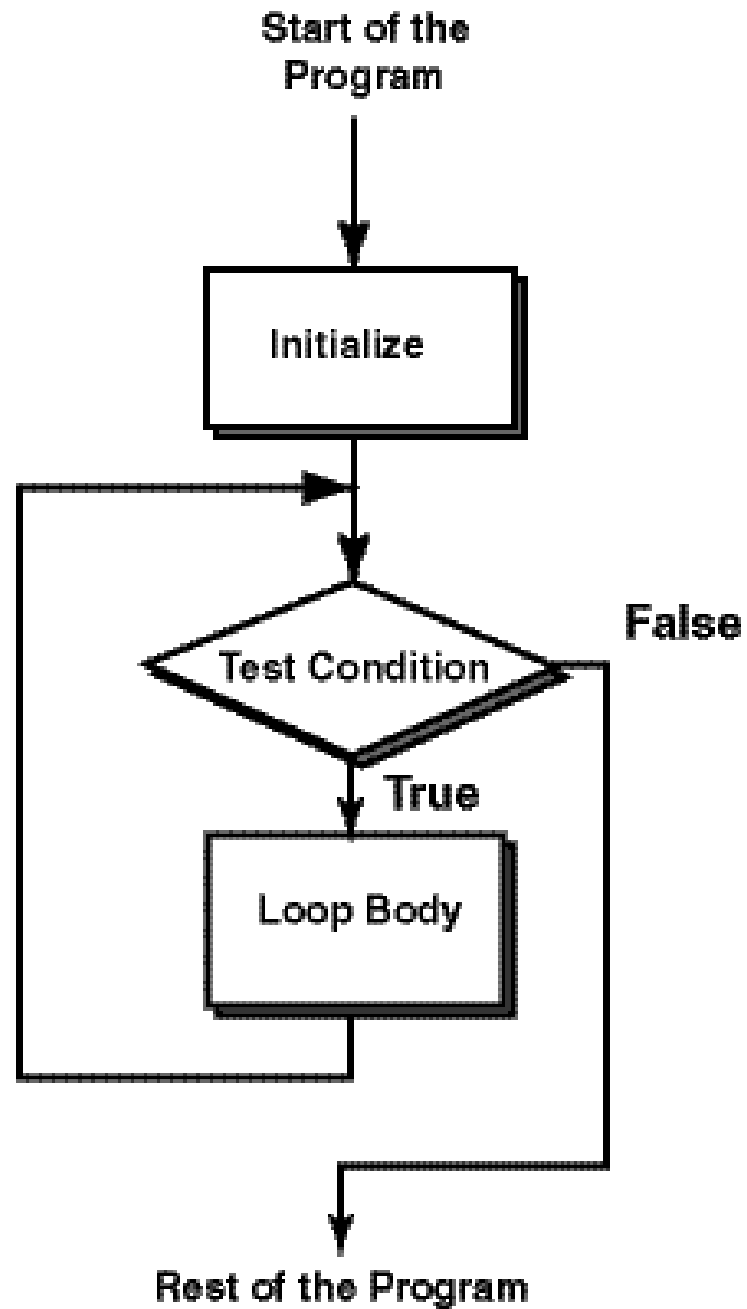
General Format:

```
...  
while <condition>  
{  
    step 1:    operation  
    ...  
    step i: operation  
}  
...
```

Execution

1. Evaluate the condition
2. If condition is true, execute steps 1 to i, then go back to 1.
3. Otherwise, if condition is false continue the execution after the while loop.

Flowchart for while



How to write a while-Loop?

1. **Formulate the test** which tells you whether the loop needs to be run again
 - `count <= 3`
2. **Formulate the actions** for the loop body which take you one step closer to termination
 - ```
{
 print("count is:")
 print(count)
 set count to count + 1 // add one to count
}
```
3. In general, **initialization** is required before the loop and some **postprocessing** after the loop
  - `set count to 1`

# Iterative Operations: Example I

---

- **Example:** Given is a natural number  $n$ . Compute the sum of numbers from 1 to  $n$ .

```
get n
set result to 0
set i to 1
while (i <= n)
{
 set result to (result+i)
 set i to (i+1)
}
print result
```

## Iterative Operations: Example II

---

**Example:** Write an algorithm that, given a positive number  $n$ , will calculate and print the value of  $n! = n \times (n - 1) \times (n - 2) \times \dots \times 1$

```
get n
set result to 1
while (n > 1)
{
 set result to (result * n)
 set n to (n - 1)
}
print result
```

# Iterative Operations: Example III

---

## Investment with Compound Interest:

Invest 10000 Euro with 5% interest compounded annually:

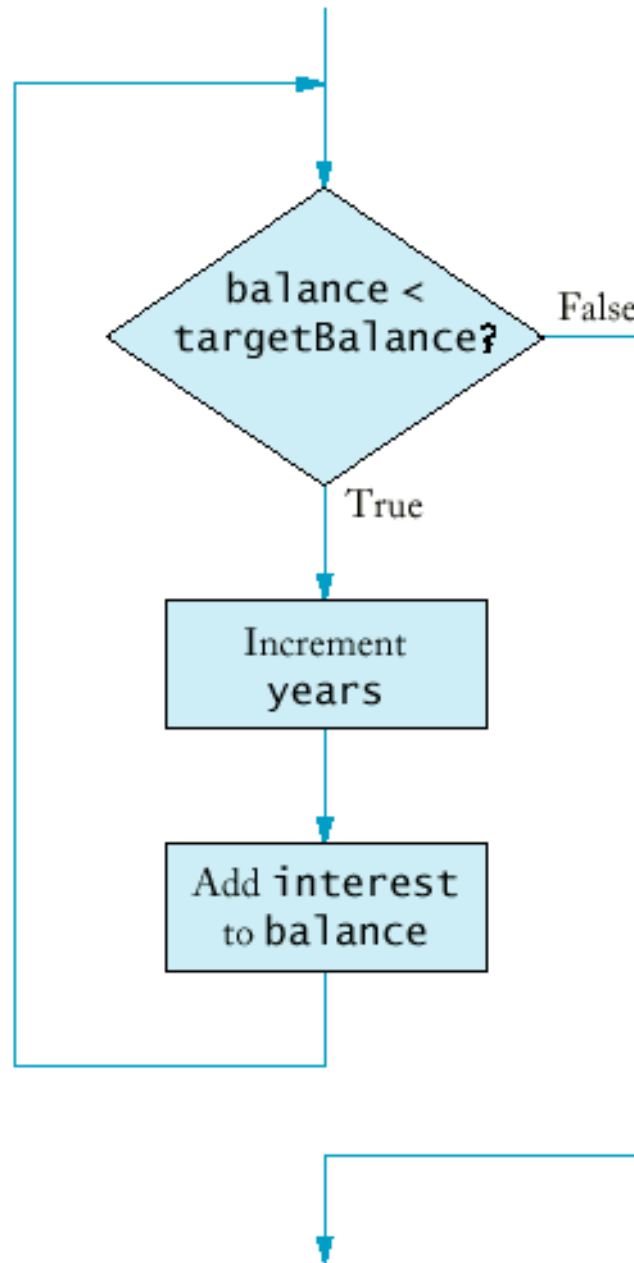
| Year | Balance   |
|------|-----------|
| 0    | 10 000    |
| 1    | 10 500    |
| 2    | 11025     |
| 3    | 11 576.25 |
| 4    | 12 155.06 |
| 5    | 12 762.82 |

**Question:** When will the balance be at least 20000 Euro?



## Flowchart for Example III

---



## Pseudo-code for Example III

---

```
get balance
get rate
set targetBalance to 20000
set year to 0
while (balance < targetBalance)
{
 set year to year + 1
 set interest to balance * rate / 100
 set balance to balance + interest
}
print("The investment doubled after")
print(year)
print("years")
```

## Infinite Loops

- **Example 1:**

```
while (3 > 2) {
 operations
}
```

- **Example 2:**

```
while (x > 0)
{
 set y to y + 1
}
```

If this loop is entered at all, it will run forever ...

# Infinite Loops

---

Why do these two algorithms not terminate?

- set the value of `i` to 1  
  while (`i < 10`)  
    {  
      print value of `i`  
    }
- set the value of `A` to 1  
  while (`A` is an odd number)  
    {  
      set `A` to `A+2`  
      print the value of `A`  
    }

# Loops: Common Errors

---

## Off-by-One Errors

- Occur when loop executes one too many or too few times
- **Example:** Add even integers from 2 to `number`, inclusive

```
...
set count to 2
set result to 0
while (count < number) {
 set result to result + count
 set count to count + 2
}
```

- Produces incorrect result if `number` is assigned an even number. Values from 2 to `number-2` will be added (i.e. `number` is excluded)
- Should be

```
...
while (count <= number)
...

```

**ALWAYS HAND SIMULATE** first, last and typical case through a loop

- to avoid off-by-one or infinite loop errors and
- to check the correctness of your algorithm.