Iterative Control Structures
Repetition / Loops

while, do-while, for

Introduction to Programming

Control Structures

Control Structures determine the order in which statements or instructions are carried out in a Java program.

The three main control structures in any programming language are

- Sequence
- Selection
- Repetition/Iteration

Control Structures

- Sequence
- Selection
  - if
  - if/else
  - switch
- Repetition/Iteration
  - while
  - do/while
  - for

3

Iteration

- So far we have looked at sequence and selection as forms of program control
- Iteration or repetition is another form of program control that allows us to instruct the computer to carry out a task over and over again by repeating a section of code
- The programming structure that is used to control this repetition is called a loop
- Loops allow us to write very powerful programs

4

Two types of repetition

- Number of repetitions is ***known***
  - *for* loop
- Number of repetitions is ***unknown***
  - *while* and *do-while* loop

Number of repetitions is ***unknown***

Examples in real life:

While there are more chips on my plate
    Eat a chip

While there is another exam to mark
    Mark the exam
    Enter mark into spreadsheet

Number of repetitions is ***unknown***

Examples in java programs:

While the user has entered an invalid mark
    Ask the user to re-enter the mark

While the user has guessed incorrectly
    Ask the user to enter another guess
    If guess is correct
        Display “well done” message

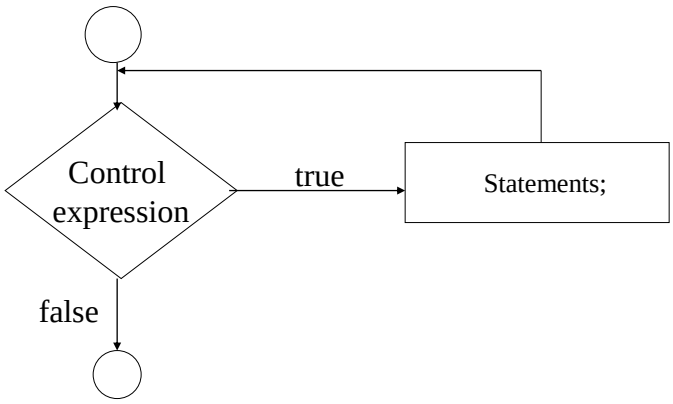
while loop

Syntax or General form

```
while(control expression)
{
     statement;
}
```

- Causes one or more statements to repeat as long as a specified expression remains ***true***
- Repetition terminates when the condition becomes ***false***
- The loop must include some feature that will alter the value of the expression thus allowing the loop to terminate

while Flowchart



while loop

1. Evaluate the control expression
2. While the control expression is true, execute the statements in the loop and go back to step 1.
3. When the control expression is false, exit the loop and execute the next statement after the loop.

Using a loop to validate data

Pseudocode

```
GET mark
WHILE mark is invalid
        GET mark
```

Number of repetitions is unknown

Read in numbers until user enters a negative value

- Pseudocode

```
Set number to 1   (or any positive value)
WHILE number is greater than 0
    GET another number
```

Number of repetitions is unknown

The `while` loop

Example:

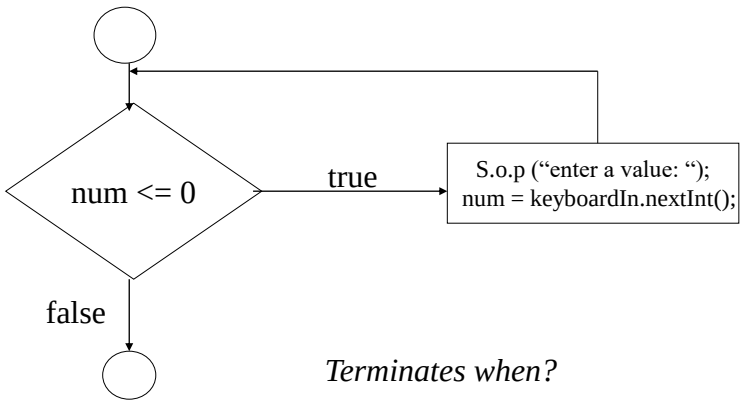
```
int number = 1;

while(number >= 0)
{
    System.out.println("Enter a number");
    number = keyboardIn.nextInt();
}
```

Number of repetitions is unknown

13

```
while (num <= 0)
{
        System.out.print("enter a value: ");
        num = keyboardIn.nextInt();
}
```



Terminates when?

Infinite loop

- A loop that runs forever
- Can be stopped by killing the program

Consider the code….

```
System.out.println("Enter a number");
number = keyboardIn.nextInt();

while(number >= 0)
{
    System.out.println("Enter a number");
}
```

Number of repetitions is unknown
When will loop terminate?

16