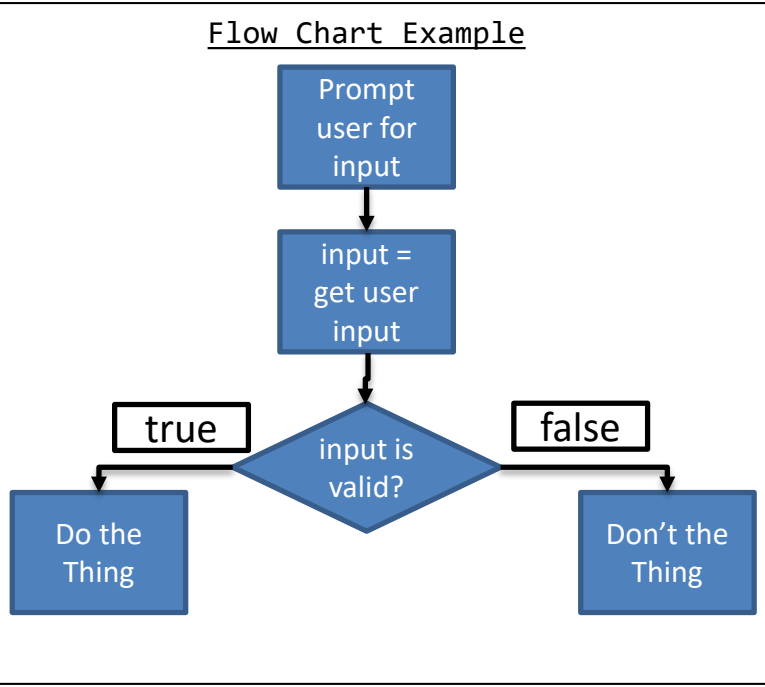
File Edit View Help

File Edit View Help

Loops
Part 01

Flow Control

- Flow of control is the order in which a program performs actions.
- A **branching statement** chooses between two or more possible actions.
- A **loop statement** repeats an action until a stopping condition occurs.
- Flow Charts diagram the flow of a program
  - Boxes are Statements
  - Diamonds are Decisions
    - True branch
    - False branch
  - Arrows indicate the flow of statements and decisions
  - Pseudocode is mostly used



Looping Statements

- While-statement
- If the Boolean expression is “true” then the body of the while-statement is executed until it is false
- Putting curly braces “{}” to denote the body of the while-statement is strongly encouraged
- Do not put a semicolon “;” after the parenthesis
  - It will ignore the Boolean expression
- Spoken
  - “while this is true then keeping doing that”

Syntax

```
while(<<Boolean expression>>)
{
      //Body of the while-statement
}
//Outside Body of the while-statement
```

Examples

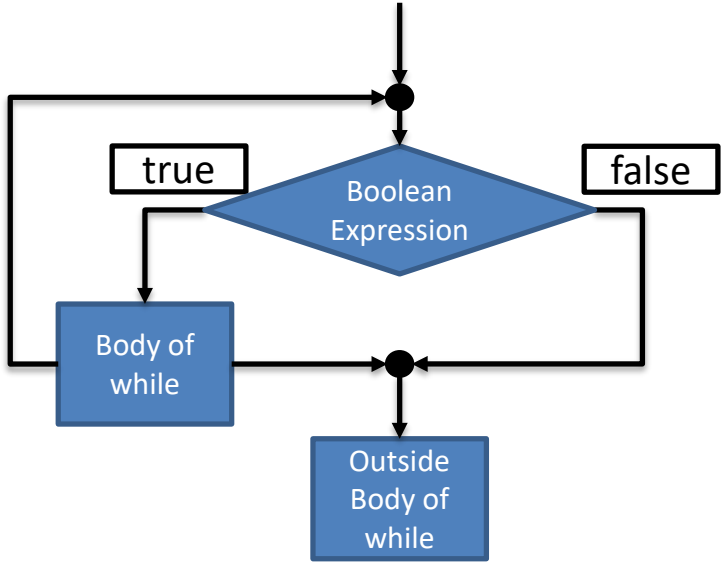
```
int a = 0;
while(a < 10)
{
      System.out.println(a);
      a++;
}
```

Looping Statements

Syntax

```
while(<<Boolean expression>>)
{
    //Body of the while-statement
}
//Outside Body of the while-statement
```

General While-Statement Flow Chart



File Edit View Help

Example

Looping Statements

- Do-while-statement
- The body of a do-while runs at least once
  - The body of a while may never run at all
- After running the body of the do-while, If the Boolean expression is “true” then the body of the do-while-statement is executed until it is false
- Putting curly braces “{}” to denote the body of the while-statement is strongly encouraged
- Put a semicolon “;” after the parenthesis
  - Otherwise it is a syntax error
- Spoken
  - “do that while this is true”

Syntax

```
do
{
      //Body of the do-while-statement
}while(<<Boolean expression>>);
//Outside Body of the do-while-statement
```

Examples

```
int a = 10;
do
{
      System.out.println(a);
      a++;
}while(a < 10);//Yes put the semicolon here
```

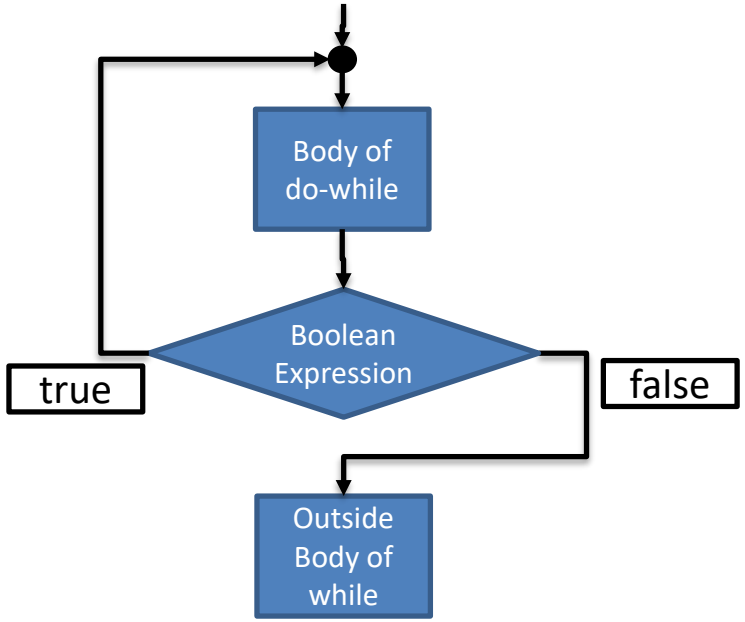
Looping Statements

Syntax

```
do
{
    //Body of the do-while
}while(<<Boolean expression>>);

//Outside Body of the do-while
```

General Do-While-Statement Flow Chart



File Edit View Help

Example

Infinite Loops

- Loop's Boolean expressions must eventually evaluate to "false"
- If this does not happen it creates a logic error called an "Infinite Loop"
- The body of the loop keeps running until the program is terminated
- Common Causes
  - Off by one errors
  - Incorrect bounds
  - Round off Errors
- Floating point types (float and double) should use "<=" or ">=" instead of "=="

Example

```
int a = 10;
while(a > 0)
{
      System.out.println(a);
      a++;
}
//Another Example
double j = 10.0;
while(j != 0.0)
{
      j -= 0.1;
      System.out.println(j);
}
```

Nested Loops

- Loops can be nested within the body of another loop
  - Much like branching statements
- Loops looping other loops can be full of logic errors

Syntax

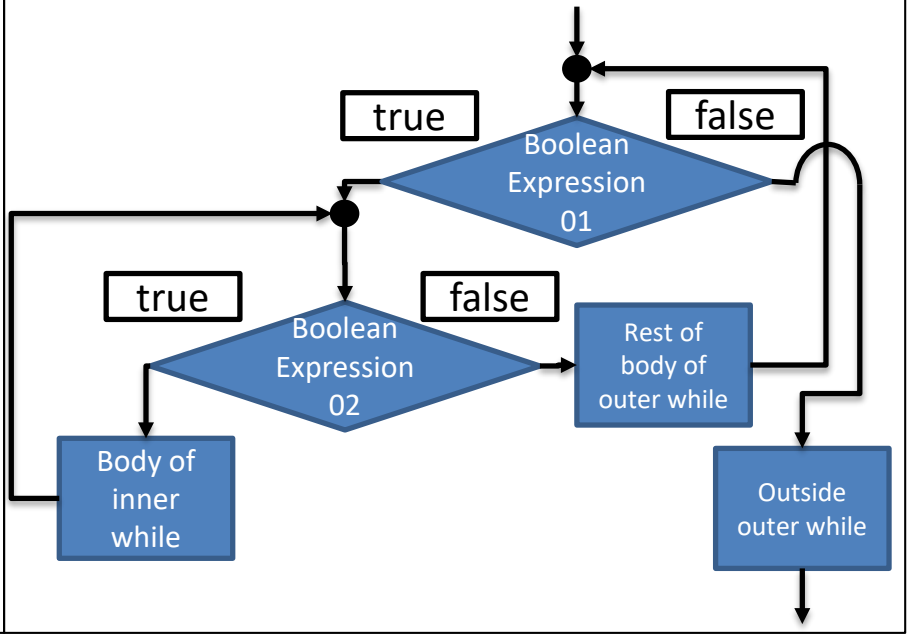
```
while(<<Boolean expression 01>>)
{
     while(<<Boolean expression 02>>)
     {
          …
     }
}
//Do-while can also be substituted
```

Looping Statements

Syntax

```
while(<<Boolean expression 01>>)
{
     while(<<Boolean expression 02>>)
     {
          //Body of inner while
     }
     //Rest of body of outer while
}
//Outside outer while
```

Nested Loop Flow Chart



File Edit View Help

Example