Variables, Operators, Conditionals and Loops

TU/e Technische Universiteit Eindhoven University of Technology

Where innovation starts

Content

- Variables
- Operators
- Conditionals
- Loops

Variables

- A variable is a *typed* and *named* storage location
  - `<type> <name> [= <value>];`

- Simple types
  - **`byte`** (-128, 127)
  - **`int`** (-10, 2147483647)
  - **`float`** (3.1425, 2.15)
  - **`char`** (´a´,´!´)
  - **`boolean`** (true, false)
  - **`String`** ("this is a string")

Variable examples

```
int myAge;
```
←Declaration

```
myAge = 40;
```
←Initialization

```
String myLanguage = "Processing";
```
←Declaration AND initialization

```
boolean isALanguage = true;
char myChar = ´a´;
float myFloat = 3.01;
int yourAge = myAge;
```
←Initializing using another variable's value

Note

- Give variables meaningful names
- Initialize variables before use
- Adhere to naming conventions (camelBack notation)
  - **`isABoy`** ok
  - **`IsABoy`** wrong (well…sort of…)
- Size matters (or in this case, case)
  - **`myVariable`** **IS NOT** **`myvariable`**

Operators

- Operators perform transformations on variables
- `=`
- `+, -, *, /, %,`
- `+=, -=, *=, /=, %=`
- `> , >= , <, <=, ==, !=`
- `&&, ||`
- `++, --`
- `(?:)`
- `()` for precedence

Operator examples (non exhaustive)

- `int x = 12;`
- `int y = 6;`
- `int xDivY = x / y;`
- `boolean xDivYIsTwo = (xDivY == 2);`
- `x++; y = x++; y = --x; y *= 4;`
- `x = x - y; y = y + x; x = y – x;`
- `float temp = 98.2;`
- `temp = temp % 5;`
- `x = (y > 6) ? 2 : 1;`
- `temp = x` (allowed… no precision loss)
- `x = temp` (not allowed…precision loss)

Conditionals

- ```
  if (<boolean condition>) {
      [<statement>;]*
  } else {
      [<statement>;]*
  }
  ```

- ```
  switch (<variable>) {
  [case <value>:[<statement>;]*]*
  [default:[<statements>;]*]
  }
  ```

if () {} else {} example

```
String val;
int x = 5;

if (x == 5) { val = "five"; }
else { val = "not five“; }

println("x = " + val);

println("x = " + ((x == 5) ? "five" : “not five"));
```

switch () {} example

```
int x = 5; String val;
switch (x) {
case 0:
  val = "zero";
  break;
case 5:
case 6:
  val = "five or six";
default:
  val = "unknown";
 break;
}
```

Do’s and don’ts

```
int x = 0;
if (x = 0) { println("x is zero"); }
else { println("x is not zero"); }
```

= != ==

```
if (x == 0) {
   println("x is zero");
} else {
   println("x is not zero");
}
```

← Use indentation

Loops (for repetitive actions)

- for (<start>; <condition>; <action>) {
  [<statement>;]*
  }
- while (<condition>) {
  [<statement>;]*
  }
- do {
  [<statement>;]*
  } while (<condition>);

for (;;) loop examples

- for (<start>; <condition>; <action>) {<statements>;}

```
int i;
for (i=0; i<5; i++) print(i);
println();

int  j;
for (i=0,j=5; i<=5; i++,j--) {
  println(i);
  println(j);
}
```

while () {} loop examples

- while (<condition>) {<statements>;}

```
int i = 0;
while (i<5) {
  i++;
  println(i);
}
while (i<5) {
  i++;
  println(i);
}
```

do {} while () loop examples

- do {<statements>;} while (<condition>);

```
int i = 0;
do {
  i++;
  println(i);
} while (i < 5);
do {
  i++;
  println(i);
} while (i < 5);
```

Homework

- Read Chapter 4 , 5 , 6 of “Learning Processing”.
- Make the exercises as pointed out in the Tasks file.
- Hand in Exercise 7.