Variables and Assignments

See also King chapter 2 - for CSC 100 Intro to Programming in C/C++

README

- This file accompanies the lecture on variables and assignments in C. To gain practice, you should type along in your own Org-mode file.

- To make this easier, use the auto expansion (`<s`), add the following two lines at the top of your file, and activate each line with `C-c C-c` (you should get a confirmation in the minibuffer):

  ```
  #+PROPERTY: header-args:C :main yes
  #+PROPERTY: header-args:C :includes <stdio.h>
  ```

- Remember that `C-M-\` inside a code block indents syntactically

Variable types and declarations

- C computes using placeholders, or **variables**
- Each variable must have a **type** to specify the data it can hold
- E.g. `int` (integer), `float` (floating point), `char` (character)
- Variables must be **declared** before they can be used, see 1

```
int height;
float profit;
char name;
```

- Several variables of the same type can be declared together:

```
int height, length, width, volume;
float profit, loss;
char first_name, last_name;
```

- Variable type declarations must precede statements that use the variables
- The block with **declarations** comes before the statements[1]

Variable assignment

- A variable gets its value through **assignment**

- In 1, the variable `height` gets the value `8`[2]

  ```
  height = 8;
  ```

- [ ] If you tried to run 1, you got an error. Can you see why?[3]
- [ ]

  However, 1 throws another error. What's wrong this time?[4]

  ```
  height = 8;
  int height;
  ```

- Code block 1 works.

  ```
  int height;
  height = 8;
  ```

- A constant assigned to a `float` variable contains a decimal point and the letter `f`, as shown in 1.

  ```
  float profit;
  profit = 2150.48f;
  ```

- Assigning a `float` to an `int` is possible but not safe

- Variables with values can be used to compute other values, as shown in 1.

  C

  ```
  int height, length, width, volume;

  height = 8;
  length = 12;
  width =10;
  volume = height * length * width;
  ```

  - To print these variables, we need to learn **formatting**

Formatting printout

- We use the built-in (via `stdio.h`) function `printf` to print

- In the code 1, `%d` is a placeholder for an `int`:

  ```
  int height;  // type declaration
  height = 8;  // variable assignment

  printf("The height is: %d\n", height); // formatted printout
  ```

  ```
  The height is: 8
  ```

- In 1, `%f` is used to print a `float`:

  ```
  float profit;       // type declaration
  profit = 2150.48f;  // variable assignment
  ```

```
printf("The profit is: $%f\n", profit); // formatted printout
```

```
The profit is: $2150.479980
```

- By default, `%f` displays the result with six digits. To change it to `p` digits, put `.p` between `%` and `f`. E.g. to print it with 2 digits, `p=2`:

```
float profit;       // type declaration
profit = 2150.48f;  // variable assignment

printf("The profit is: $%.2f\n", profit); // formatted printout
```

```
The profit is: $2150.48
```

Glossary

TERM	EXPLANATION
Variable	
Type	
Type declaration	
`int`	
`float`	
`char`	
Formatting	
`%d`	
`%f`	

Footnotes:

[1] In the C99 standard, declarations don't have to come before statements.

[2] The value `8` is called a `constant` because it cannot change

[3] Assignment is variable use. Variable types must be declared before they can be used.

[4] The declaration must precede the use of the variable.

Author: Marcus Birkenkrahe
Created: 2022-02-07 Mon 10:31
Validate