

COMP1917: 01_Numbers In, Numbers Out

Sim Mautner

simm@cse.unsw.edu.au

July 26, 2016

References

- Moffat, Chapter 2.

Variables and Types

- Variables are used to store data. → boxes
- Each variable has a type. → size/structure of the box
- For now, we are using 3 data types:

char	character	'A', 'e', '#'
int	integer	2, 17, -5
float	floating point number	3.14159, 2.71828

Variables

- Declare
The first time a variable is mentioned, we need to specify its type.
- Initialise
Before using a variable we need to assign it a value.
- Assign
To give a variable a value.

```
int num; // Declare
num = 5; // Initialise (also Assign)
...
num = 27; // Assign
```

Variables

- We can also Declare and Initialise in the same step:

```
int num = 5; // Declare and Initialise  
...  
num = 27; // Assign
```

Variable Names (and other Identifiers)

- Must be made up of letters, digits and underscores ('_')
- The first character must be a letter
- Are case sensitive (num1 and Num1 are *different*)
- **Restrictions:** Keywords like:
 if, while, do, int, char, float
cannot be used as identifiers

Output using printf()

- No variables:

```
printf("Hello World\n");
```

- A single variable:

```
int num = 5;  
printf("num is %d\n", num);
```

- More than one variable:

```
int num1 = 5;  
int num2 = 17;  
printf("num1 is %d and num2 is %d\n", num1, num2);
```

- **Note:** The order in which the variables are listed, is the order in which they will appear.

```
int num1 = 5;  
int num2 = 17;  
printf("num2 is %d and num1 is %d\n", num2, num1);
```

Output using printf()

Placeholders:

- char uses %c
- int uses %d
- float uses %f
- double uses %lf

Try It Yourself:

- 1 Copy the code from the end of the previous slide into a C program and run it.
- 2 Make the appropriate changes so that it declares, initialises and prints a char, float and double.

Input using scanf()

- Example:

```
int num = 0;
scanf("%d\n", &num);
printf("num = %d\n", num);
```

- Notice that the variable is still initialised. (Not necessary, but good practice.)
- Notice the & before the variable name. Don't forget it!!

Input using scanf()

- Multiple variables (space separated):

```
int num1 = 0;
int num2 = 0;
scanf("%d %d\n", &num1, &num2);
printf("num1 = %d and num2 = %d\n", num1, num2);
```

- Multiple variables (comma separated):

```
int num1 = 0;
int num2 = 0;
scanf("%d, %d\n", &num1, &num2);
printf("num1 = %d and num2 = %d\n", num1, num2);
```

- Notice the space or comma between the variables.

Input using `scanf()`

Try It Yourself:

- Create a C program using the code from the previous slide.
- Using what you know about placeholders for `printf()` (earlier this lecture) and `scanf()`, make the changes required so that it scans in and prints out a character (`char`).

Programming Task

Write a program to:

- 1 Read in a number.
- 2 Compute the cube of that integer.
- 3 Display the result on the screen.

Programming Task

Process:

- ① Step 1: Think about the problem.
- ② Step 2: Break it down into steps (and each step into smaller steps).
- ③ Step 3: Convert the basic steps into code.
- ④ Step 4: Compile the program.
- ⑤ Step 5: Test the program on a range of data.

Arithmetic Operators

Name	Symbol	Example	Conditions
Add	+	$a + b$	none
Subtract	-	$a - b$	none
Multiply	*	$a * b$	none
Divide	/	a / b	ignores remainder for integer division
Modulus	%	$a \% b$	remainder of a/b

```
int result = 50 / 3;
int remainder = 50 % 3;
printf("50 divided by 3 equals %d remainder %d\n",
       result, remainder);
```

printf() with floats

<code>%d</code>	decimal integer
<code>%5d</code>	decimal integer at least 5 chars wide
<code>%f</code>	floating point number
<code>%5f</code>	floating point number at least 5 chars wide
<code>%.3f</code>	floating point number 3 decimal places
<code>%5.3f</code>	floating point number at least 5 chars 3 decimal places

Math Equations

- `sqrt()`, `sin()`, `cos()`, `log()`, `exp()`
- `#include <math.h>`
- Compile with `-lm`

Math Equations: Example

```
#include <stdio.h>
#include <math.h>

int main(int argc, char *argv[]) {

    int num = 0;
    scanf("%d", num);
    double result = sqrt(num);
    printf("The square root is: %.3lf\n");
    return 0;
}
```