

# Types and Operators

Arash Rafiey

August 24, 2017

We are given a number; the aim is to calculate its square.  
For example, 16 squared = 256.

**Abstraction:** 16 can be represented using a variable.

```
int n = 16;
```

```
int square = n * n;
```

# Variables

We are given a number; the aim is to calculate its square.  
For example, 16 squared = 256.

**Abstraction:** 16 can be represented using a variable.

```
int n = 16;
```

```
int square = n * n;
```

A **variable** is:

- 1 a portion of the computer's memory, storing a **value**,
- 2 plus associated name, called an **identifier**.

# Identifiers

An **identifier** is a sequence of characters to name:

- ① Variable, label.
- ② Functions.
- ③ structure, enum (enumeration), union.
- ④ Members of a structure, union, or enumeration.

An **identifier** is a sequence of characters to name:

- 1 Variable, label.
- 2 Functions.
- 3 structure, enum (enumeration), union.
- 4 Members of a structure, union, or enumeration.

RegEx for C identifier: `[_a-zA-Z][_a-zA-Z0-9]{0,30}`

Identifiers are **case-sensitive**: lowercase and uppercase letters are considered different.

Don't begin variable names with the **underscore** because standard library routines often use such names.

**Basic C types** are usually supported by hardware.

## ① Integers:

- ① `char` is a character (1 byte)
- ② `short` (at least 2 bytes)
- ③ `int` (at least 2 bytes, 4 bytes in modern compilers)
- ④ `long` (at least 4 bytes, 8 bytes in Linux)
- ⑤ `long long` (at least 8 bytes)

## ② Floating point numbers:

- ① `float` (4 bytes)
- ② `double` (8 bytes)
- ③ `long double` (10 bytes)

Size and un/signed is compiler/machine-dependent.

**Signed/unsigned** integers:

- ① `unsigned char` (0..255), `signed char` (-128..127)
- ② `signed long int` is typically a synonym for `long`

# Constant literals

An integer constant like 123 is `int` by default.

An floating point number 1.5 has `double` type.

All store ASCII code of 'A', type `char`: 'A', has ASCII code 65.

**Suffixes** are used to specify the type:

- 1 `long` 123456789L
- 2 `unsigned long` 0x789ABCUL (hexadecimal)
- 3 `unsigned long` 0XABCDE0123456789ULL (hexadecimal)
- 4 `unsigned int` 1234u
- 5 `float` 1e-2f
- 6 `long double` 1e-2L

- 1 **Arithmetic:**
  - + and - have lower precedence than \*, /, %  
// integer remainder
  - Increment/decrement prefix/postfix: ++a, --a, a++, a--
- 2 **Comparison/relational:** ==, !=, <, >, <=, >=
- 3 **Logical:** ! not, && and, || or
- 4 **Bitwise:** ~ (NOT), & (AND), |(OR), <<, >>
- 5 **Compound assignment:** a+ = b; // it adds b to a
- 6 **Compound assignment:** a\* = b; // it multiplies a with b



# Increment/decrement Demo

```
int a = 10, b = (5 - 4 * -4) % 11, c = (10 + 11) / 2, d = (8 + 13) >>1;
```

# Increment/decrement Demo

```
int a = 10, b = (5 - 4 * -4) % 11, c = (10 + 11) / 2, d = (8 + 13) >>1;
```

```
printf(" ++a=%d, --b=%d, c++=%d, d--=%d",  
 ++a, --b, c++, d--);
```

# Increment/decrement Demo

```
int a = 10, b = (5 - 4 * -4) % 11, c = (10 + 11) / 2, d = (8 + 13) >>1;
```

```
printf(" ++a=%d, --b=%d, c++=%d, d--=%d",  
 ++a, --b, c++, d--);
```

The **output** is:

```
++a=11, --b=9, c++=10, d--=10
```

# Increment/decrement Demo

```
int a = 10, b = (5 - 4 * -4) % 11, c = (10 + 11) / 2, d = (8 + 13) >>1;
```

```
printf(" ++a=%d, --b=%d, c++=%d, d--=%d",  
 ++a, --b, c++, d--);
```

The **output** is:

```
 ++a=11, --b=9, c++=10, d--=10
```

```
printf(" Final a=%d, b=%d, c=%d, d=%d", a, b, c, d);
```

# Increment/decrement Demo

```
int a = 10, b = (5 - 4 * -4) % 11, c = (10 + 11) / 2, d = (8 + 13) >>1;
```

```
printf(" ++a=%d, --b=%d, c++=%d, d--=%d",  
 ++a, --b, c++, d--);
```

The **output** is:

```
 ++a=11, --b=9, c++=10, d--=10
```

```
printf(" Final a=%d, b=%d, c=%d, d=%d", a, b, c, d);
```

```
Final a=11, b=9, c=11, d=9
```

# Swap two numbers puzzle

```
int a = 10, b = 8;
```

How to **swap two numbers**?

# Swap two numbers puzzle answers

```
int a = 10, b = 8;
```

How to **swap two numbers**?

```
int temp = a;
```

```
a = b;
```

```
b = temp;
```

# Swap two numbers puzzle answers

```
int a = 10, b = 8;
```

How to **swap two numbers**?

```
int temp = a;
```

```
a = b;
```

```
b = temp;
```

.....Second Solution.....

```
a = a + b;
```

```
b = a - b;
```

```
a = a - b;
```



# Swap two numbers puzzle answers

```
int a = 10, b = 8;
```

How to **swap two numbers**?

```
int temp = a;
```

```
a = b;
```

```
b = temp;
```

.....Second Solution.....

```
a = a + b;
```

```
b = a - b;
```

```
a = a - b;
```

.....Third Solution.....

```
a = a ^ b; // it does bits wise xor
```

```
b = a ^ b;
```

```
a = a ^ b ;
```