C++ Variables

Instructor: Chi Tse (Ricky)

Variables in C++

- Allows storage of data internally for the program
- Allows storage of information from the user
- There are different types of variables which services different needs

Examples:
- Variables that store texts
- Variables that store integers (positive & negative numbers)
- Variables that store decimals (floating point numbers)

Declare / Create Variables

In order to use a variable in C++, we must first declare (i.e. create) it.

Model:

variable_type   variable_name ;

– variable_type: The type of a variable, depends on the type of data we want to store.

– variable_name: The name of a variable, how we like to call it in the rest of the program.

NAME	VALUE	TYPE
number	123	int
sum	-456	int
pi	3.1416	double
average	-55.66	double

A variable has a *name*, stores a *value* of the declared *type*

Variable Naming Convention

- C++ rules for legal variable names:
  1. Cannot start with a number. Should start with a letter.
  2. The rest of the name can be a letter, a number, or an underscore "_". (i.e. no space or special characters)
  3. Cannot be a C++ keyword.
- Suggestions for variable names
  - Should be meaningful
  - Should be easy to read
- Check p.7 of the textbook for more information

Illegal Variable Names

Example of illegal variable names

- int number of students;
  - Has spaces
- int 1number;
  - Begins with a number
- int discount%;
  - Contains a symbol
- double int;
  - Contains a keyword, int

Important Note

- C++ is case sensitive!
- Examples:

  int hello; //declares a variable hello

  int Hello; //declares another variable Hello

  Int hello; //error, Int is not a C++ type

  Double amount; //error, Double is not a type

Are the following **variable names** valid?

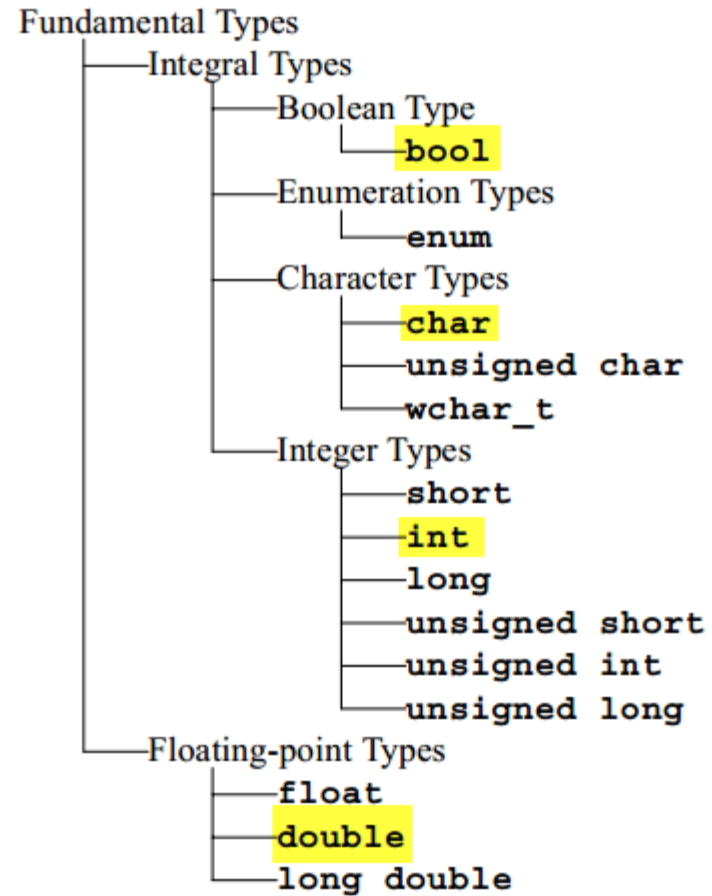|	
a) student name	b) int
c) student_name	d) 111students
e) Fall2014	f) john@cuny
g) "variable_name"	h) return
i) return0	j) _111students

Answers:

a) student name N (no spaces allowed)	b) int N (C++ keyword)
c) student_name Y	d) 111students N (can't start with a number)
e) Fall2014 Y	f) john@cuny N (no special symbols)
g) "variable_name" N (no special symbols)	h) return N (C++ keyword)
i) return0 Y	j) _111students Y

Variable Types

In CS111, we will focus on the
following variable types:
int, double, bool, char, string

```
Fundamental Types
    Integral Types
        Boolean Type
            bool
        Enumeration Types
            enum
        Character Types
            char
            unsigned char
            wchar_t
        Integer Types
            short
            int
            long
            unsigned short
            unsigned int
            unsigned long
    Floating-point Types
        float
        double
        long double
```

(from p.16 of Schaum's Outlines - Programming with C++)

Data Types Range

Table B.1: C++ Intrinsic Data Types

TYPE	DESCRIPTION (FOR 32-BIT SYSTEMS)	RANGE (ON 32-BIT SYSTEMS)
char	1-byte integer (used to hold ASCII character value)	0 to 255
unsigned char	1-byte unsigned integer	0 to 255
signed char	1-byte signed integer	-128 to 127
short	2-byte integer	-32,768 to 32,767
unsigned short	2-byte unsigned integer	0 to 65,535
int	4-byte integer (but same as short on 16-bit systems)	Approx. ± 2 billion
unsigned int	4-byte unsigned integer (but same as unsigned short on 16-bit systems)	Approx. 4 billion
long	4-byte integer	Approx. ± 2 billion
unsigned long	4-byte unsigned integer	Approx. 4 billion
bool	Integer in which all nonzero values are converted to **true** (1); also holds **false** (0) (ANSI)	**true** or **false**
wchar_t	Wide character, for holding Unicode characters (ANSI)	Same as **unsigned int**
long long	64-bit signed integer (C++0x)	Approx. $\pm 9 \times 10^{18}$
unsigned long long	64-bit unsigned integer (C++0x)	Approx. 1.8×10^{19}
float	Single-precision floating point	3.4×10^{38}
double	Double-precision floating point	1.8×10^{308}
long double	Extra-wide double-precision (ANSI)	At least as great as **double**

(from Appendix B of Brian Overland's textbook)

int (integers)

- Variable declaration
  - int number;
  - int year;
  - int age;

- Examples of VALID integer value assignment
  - number = 3;
  - year = 2016;
  - age = 20;

- Examples of INVALID integer value assignment
  - number = "3";
  - year = '2016';
  - age = "thirty";
  - age = 20.5; //truncated the value to 20

double (decimals, high precision)

- Variable declaration
  - double pi;
  - double e;
- Examples of VALID double value assignment
  - pi = 3.1415926535;
  - e = 2.71828;
- Examples of INVALID double value assignment
  - pi = "3.141";
  - pi = ' 3.141 ';

bool(boolean: true or false)

- Variable declaration
  - bool reply;
  - bool answer;

- Examples of VALID bool values
  - answer = true;
  - answer = false;
  - reply = 0; //(i.e. false)
  - reply = 1;

- Examples of INVALID bool values
  - answer = "true";
  - reply = '0'; //value becomes true

char (characters)

- Variable declaration
  - char c;
  - char newline;
  - char code;

- Examples of VALID char values
  - c = 'c';
  - newline = '\n';
  - code = 165; //Yen symbol (¥) in Unicode

- Examples of INVALID char values
  - code = 456;
  - newline = "\n";

string

- Variable declaration
  - string name;
  - string address;
  - string day;

- Examples of VALID string values
  - name = "Vincent";
  - address = "65-30 Kissena Blvd.";
  - day = "2";

- Examples of INVALID string values
  - name = ' Vincent ';
  - address = 65-30 Kissena Blvd.;
  - day = 2;