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 service different needs

Examples:
- Variables that store texts
- Variables that store integers (positive & negative numbers)
- Variables that store decimals (floating point numbers)
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.
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.26 (3rd ed) or p.29 (2nd ed) of textbook for more information
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`
• 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
<table>
<thead>
<tr>
<th>a) student name</th>
<th>b) int</th>
</tr>
</thead>
<tbody>
<tr>
<td>c) student_name</td>
<td>d) 111students</td>
</tr>
<tr>
<td>e) Fall2014</td>
<td>f) john@cuny</td>
</tr>
<tr>
<td>g) &quot;variable_name&quot;</td>
<td>h) return</td>
</tr>
<tr>
<td>i) return0</td>
<td>j) _111students</td>
</tr>
<tr>
<td>a) student name</td>
<td>b) int</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>N (no spaces allowed)</td>
<td>N (C++ keyword)</td>
</tr>
<tr>
<td>c) student_name</td>
<td>d) 111students</td>
</tr>
<tr>
<td>Y</td>
<td>N (can’t start with a number)</td>
</tr>
<tr>
<td>e) Fall2014</td>
<td>f) john@cuny</td>
</tr>
<tr>
<td>Y</td>
<td>N (no special symbols)</td>
</tr>
<tr>
<td>g) &quot;variable_name&quot;</td>
<td>h) return</td>
</tr>
<tr>
<td>N (no special symbols)</td>
<td>N (C++ keyword)</td>
</tr>
<tr>
<td>i) return0</td>
<td>j) _111students</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
In CS111, we will focus on the following variable types: int, double, char, bool, string

(from p.16 of reference textbook)
• 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;
• 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 ';
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;
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";
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