CS 161
Intro to CS I

Expressions & User Input
Chap. 1.3 – 1.5
Expressions

• What is an expression?
  – Set of operations producing a value
    • Combining simple values
      12 * 4 + 6 * 10 vs. ((12 * 4) + 6) * 10
Expressions cont.

- Pieces of an Expression:
  - Operators
    - Indicate operation, e.g. +, *, /, -, %
  - Operands
    - Values in the expression
  - Evaluation
    - Process of obtaining results from operations on operands
Arithmetic Operators

- Add
  \[ 34 + 23 \]
- Subtract
  \[ 34 - 23 \]
- Multiply
  \[ 2 * 23 \]
- Divide
  \[ 40 / 10 \]
- Remainder/Mod
  \[ 34 \% \ 5 \]
Arithmetic

• Integer Arithmetic
  
  ```cpp
  std::cout << 3/8;  /*prints 0*/
  std::cout << 34/5;  /*prints 6*/
  ```

• Floating Point Arithmetic
  
  ```cpp
  std::cout << 34.0/5.0;  /*prints 6.8*/
  std::cout << 3.0/8;  /*prints .375*/
  std::cout << 3/8.0;  /*prints .375*/
  ```
Type Casting

• Casting

```cpp
std::cout << 34 / (int) 5.0; /*prints 6*/
std::cout << (int) (34 / 5.0); /*prints 6*/
std::cout << (float) 34 / 5; /*prints 6.8*/
```

• What is wrong with these?

```cpp
std::cout << (int) 34 / 5.0; /*prints 6.8*/
std::cout << (float) (34/5); /*prints 6.0*/
```
```cpp
#include <iostream>
#include <climits>
using std::cout;
using std::endl;

int main() {
    unsigned short us_num_max;

    //This will give a compiler warning for overflow, but needs no typecast
    //us_num_max = USHRT_MAX + 1;
    cout << "Unsigned Short Max: " << us_num_max << endl;

    //This will not give a compiler warning, but needs typecast
    us_num_max = (short) USHRT_MAX;
    cout << "Unsigned Short Max: " << (short) us_num_max + 1 << endl;
    cout << "Unsigned Short Max: " << (unsigned short) (us_num_max + 1) << endl;

    return 0;
}
```
Precedence

• What is precedence?
  – Binding power of operator
  – (*, /, %) vs. (+, -)

• How do we override precedence?
  – Parenthesis!

• Examples:
  12 * 4 + 6 * 10 vs. ((12 * 4) + 6) * 10
int main() {
    //declare variables
    double height;
    double weight;
    double bmi;

    //compute BMI
    height = 70.0;
    weight = 195.0;
    bmi = weight / (height*height) * 703;

    //print results
    std::cout << “Current BMI: ” << bmi << std::endl;

    return 0;
}
New Operators for Expressions

• What if you have the statement `var = var + 1;`  
  `var += 1; /*Add operand on right to var*/`  
  `var++; /*Increment var by one*/`

• What if you have the statement `var = var - 5;`  
  OR `var = var + var;`  
  `var -= 5;`  
  `var += var;`

• Pre vs. Post increment: `++var` vs. `var++`
```cpp
#include <iostream>
#include <climits>
using std::cout;
using std::endl;

//using namespace std;

int main() {
    unsigned short us_num_max;

    //This will give a compiler warning for overflow, but needs no typecast
    //us_num_max = USHRT_MAX + 1;
    cout << "Unsigned Short Max: " << us_num_max << endl;

    //This will not give a compiler warning, but needs typecast
    us_num_max = USHRT_MAX;
    cout << "Unsigned Short Max: " << (short) us_num_max + 1 << endl;
    cout << "Unsigned Short Max: " << (unsigned short) (us_num_max + 1) << endl;

    //We can show overflow and change value of variable
    cout << "Unsigned Short Max: " << ++us_num_max << endl;
    cout << "Unsigned Short Max: " << us_num_max << endl;

    //We don't show overflow because prints before increment
    //cout << "Unsigned Short Max: " << us_num_max++ << endl;
    //cout << "Unsigned Short Max: " << us_num_max << endl;

    //We show overflow and change value of variable twice, i.e. +2
    //cout << "Unsigned Short Max: " << (++us_num_max)++ << endl;
    //cout << "Unsigned Short Max: " << us_num_max << endl;

    return 0;
}
```
How do we read into a variable in C++?

- Declare a variable
- Read value from user and store at variable location
- How do we do this?

```
#include <iostream>
int main() {
    int x;
    std::cin >> x;
    std::cout << x;
    return 0;
}
```
Reading/Assignments...

• Read Chap. 2.1