CS 161
Intro to CS I

Functions

Odds and Ends

Oregon State University
College of Engineering

- Assignment 3 code due Sunday.
- Study sessions: WITH 109, M/T 5-6:30
- Exam I next Wednesday, 2/6/2019

in class, same location, 50 min.

- Proficiency Demo Practice - lab 5
  5 pts

Decomposition

- Divide Problem (task) Into Subtasks
  - Procedural Decomposition
  - Examples: cooking, cleaning, etc.
- Incremental Programming
  - Iterative Enhancement (Stepwise Refinement)
- Examples: Replicating Code

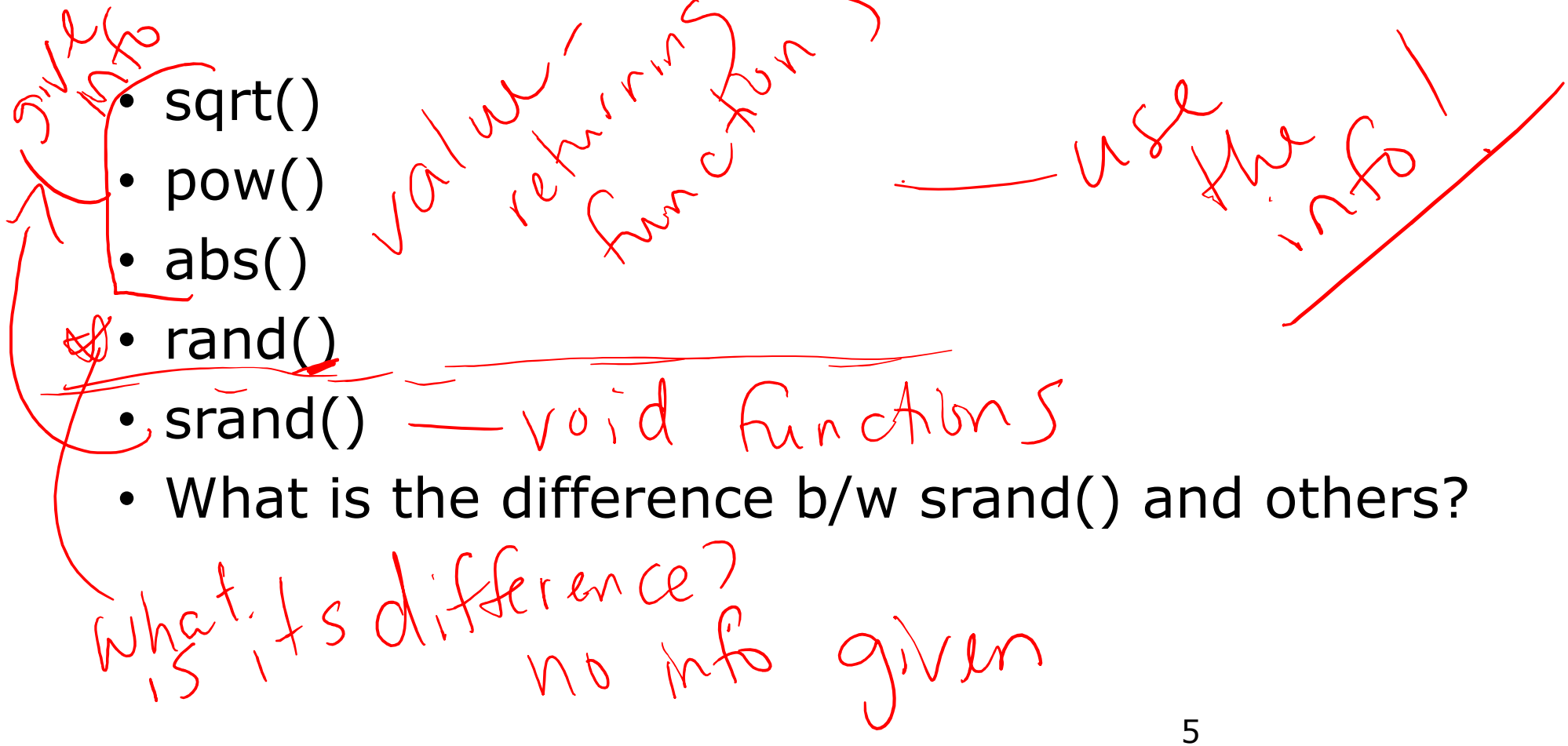
Functions

Oregon State University
College of Engineering

- What is a function?
  - Block of code to perform action/subroutine
- When have we seen functions already?
  - Predefined
- What is the purpose?
  - Reduce
  - Reuse
  - Readability

3 Rs

Predefined Functions



- sqrt()
- pow()
- abs()
- rand()
- srand()
- What is the difference b/w srand() and others?

Procedural Decomposition

- Functions
  - int **main**() { }
  - User defined

    void draw_box() { }
- Function Call
  - draw_box();

Procedural Decomposition

Oregon State University
College of Engineering

```
#include <iostream>
using std::cout;
int main() {
    cout << "+--------+\n";
    cout << "|        |\n";
    cout << "+--------+\n";
    cout << "+--------+\n";
    cout << "|        |\n";
    cout << "+--------+\n";
    return 0;
}
```

```
#include <iostream>
using std::cout;
void draw_box();  //Declare function
int main() {
    draw_box();  //Use function
    draw_box();
    return 0;
}
void draw_box() {  //Define function
    cout << "+--------+\n";
    cout << "|        |\n";
    cout << "+--------+\n";
}
```

Functions Calling Other Functions

Oregon State University
College of Engineering

```
#include <iostream>
void draw_box();
void draw_top_bottom();
void draw_sides();
int main() {
      draw_box();
      return 0;
}
void draw_box() {
      draw_top_bottom();
      draw_sides();
      draw_top_bottom();
}
void draw_top_bottom() {
      std::cout << "+--------+\n";
}
void draw_sides() {
      std::cout << "|        |\n";
}
```

general is more abstract

Generalization

Oregon State University
College of Engineering

- Does a function make a task more specific or more general?
  - Justification
  - Examples

void Functions

Oregon State University
College of Engineering

- Doesn’t return a value
- Still has arguments/parameters

Scope (Visibility)

Oregon State University
College of Engineering

- Part of program in which a declaration is valid
- Local variable
  - Declared inside a function only accessible inside function
- Localizing variables
  - Declaring variable in innermost scope

Illegal access outside loops

Oregon State University
College of Engineering

```
for(x = 0; x < 10; x++) {
      int y = 10;
      cout << "The value of x * y is: " << x*y << endl;
}
cout << "The value of y is: " << y << endl;  /*y outside scope*/
```

- How do we fix this?
- What about if/else blocks?

Illegal access in functions

Oregon State University
College of Engineering

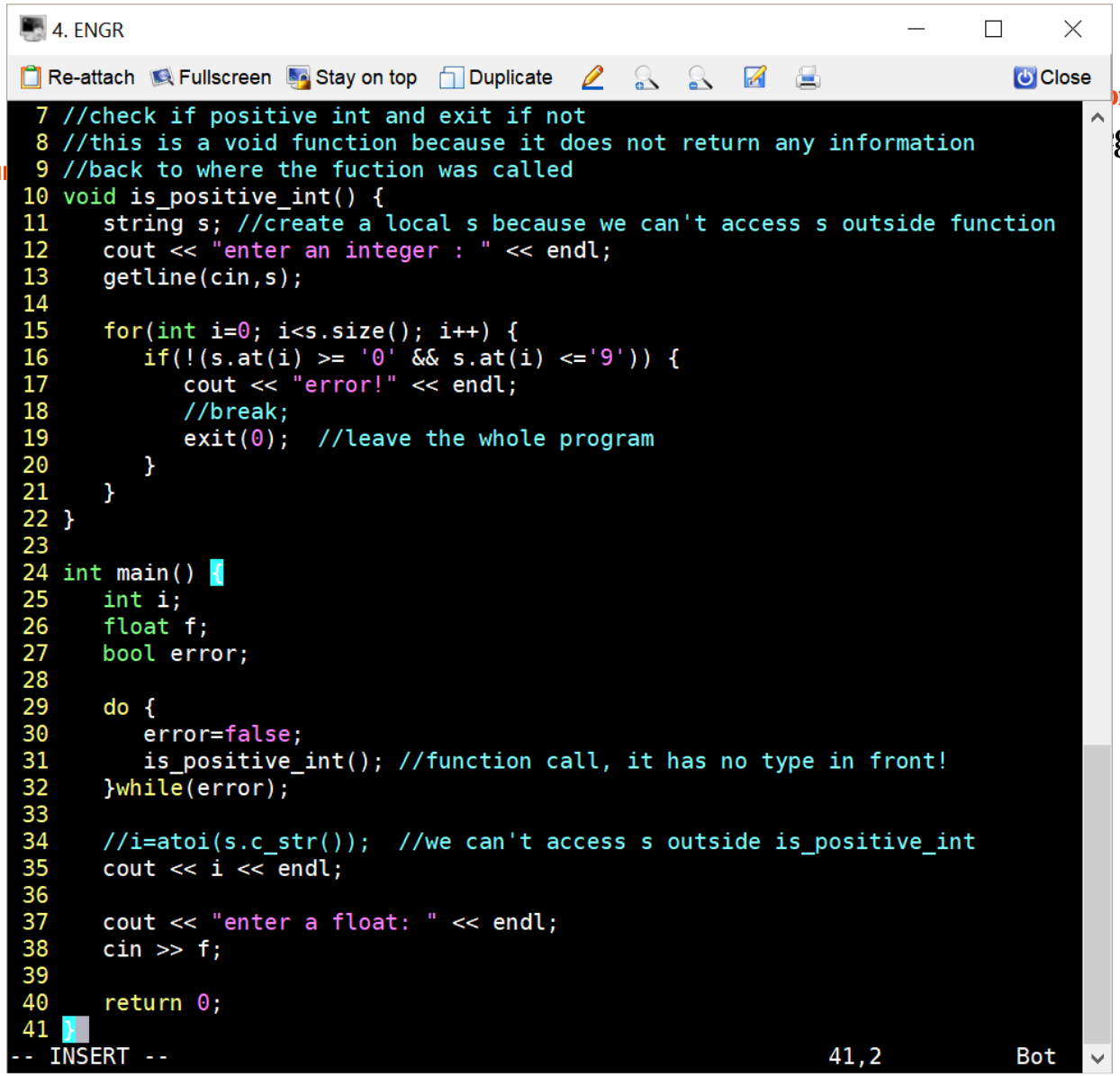
```
int main () {
      int x=2, y=3;
      compute_sum();
      sum = x+y;  //error: sum hasn't been declared
      return 0;
}
void compute_sum() {
      int sum = x+y;  //error: x and y outside scope
}
```

Arguments/Parameters Demo

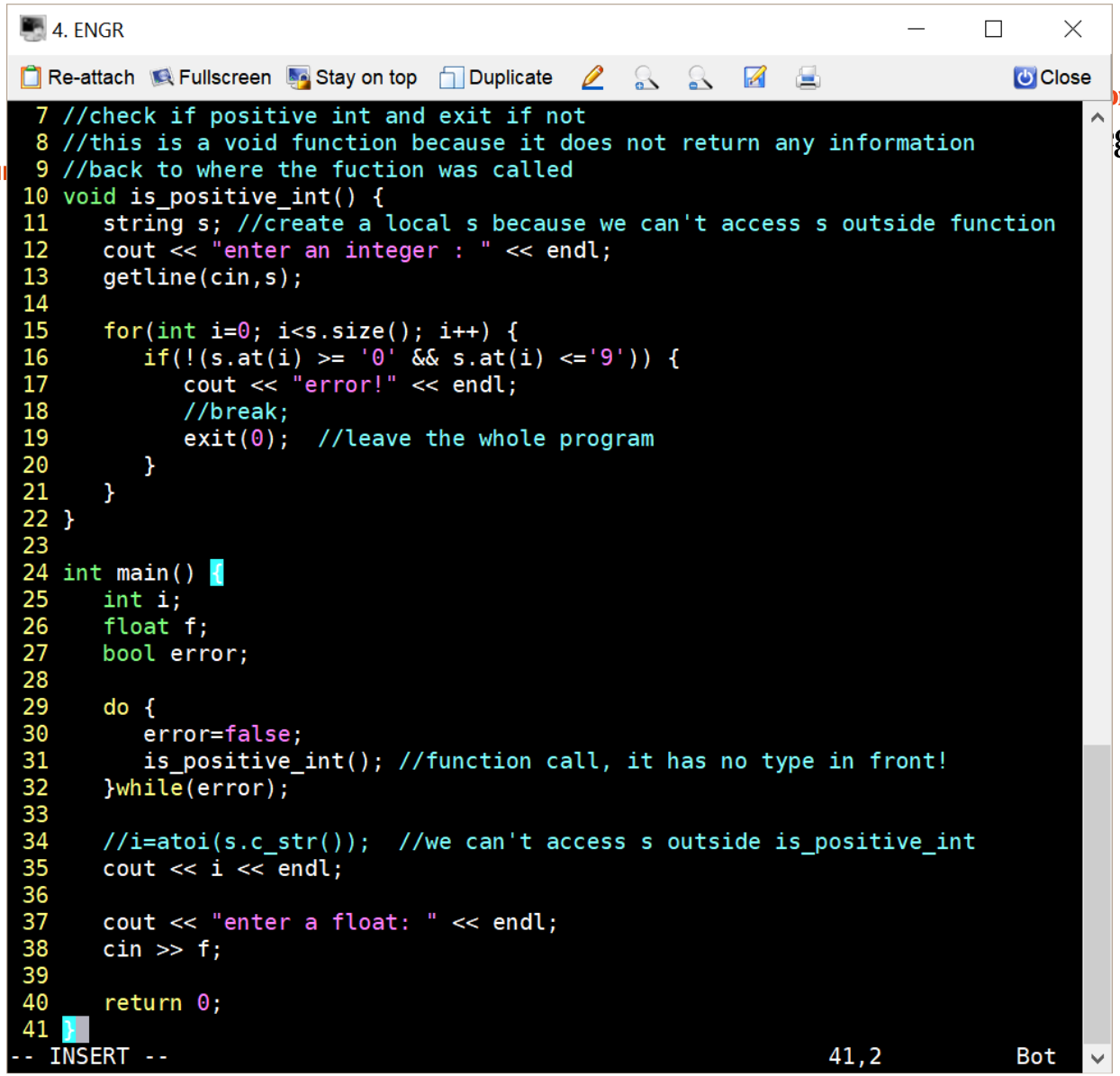
Oregon State University
College of Engineering

- How could you make an is_positive_int() function?
- How could you make an is_positive_float() function?
- Does it make sense to have it void?

Demo...


4. ENGR
Re-attach
Fullscreen
Stay on top
Duplicate
Close
7 //check if positive int and exit if not
8 //this is a void function because it does not return any information
9 //back to where the fuction was called
10 void is_positive_int() {
11 string s; //create a local s because we can't access s outside function
12 cout << "enter an integer : " << endl;
13 getline(cin,s);
14
15 for(int i=0; i<s.size(); i++) {
16 if(!(s.at(i) >= '0' && s.at(i) <='9')) {
17 cout << "error!" << endl;
18 //break;
19 exit(0); //leave the whole program
20 }
21 }
22 }
23
24 int main() {
25 int i;
26 float f;
27 bool error;
28
29 do {
30 error=false;
31 is_positive_int(); //function call, it has no type in front!
32 }while(error);
33
34 //i=atoi(s.c_str()); //we can't access s outside is_positive_int
35 cout << i << endl;
36
37 cout << "enter a float: " << endl;
38 cin >> f;
39
40 return 0;
41 }
-- INSERT -- 41,2 Bot

Demo...


4. ENGR
Re-attach
Fullscreen
Stay on top
Duplicate
Close
7 //check if positive int and exit if not
8 //this is a void function because it does not return any information
9 //back to where the fuction was called
10 void is_positive_int() {
11 string s; //create a local s because we can't access s outside function
12 cout << "enter an integer : " << endl;
13 getline(cin,s);
14
15 for(int i=0; i<s.size(); i++) {
16 if(!(s.at(i) >= '0' && s.at(i) <='9')) {
17 cout << "error!" << endl;
18 //break;
19 exit(0); //leave the whole program
20 }
21 }
22 }
23
24 int main() {
25 int i;
26 float f;
27 bool error;
28
29 do {
30 error=false;
31 is_positive_int(); //function call, it has no type in front!
32 }while(error);
33
34 //i=atoi(s.c_str()); //we can't access s outside is_positive_int
35 cout << i << endl;
36
37 cout << "enter a float: " << endl;
38 cin >> f;
39
40 return 0;
41 }
-- INSERT -- 41,2 Bot

Global Variables
Do NOT use them!!!