

CS 161

Intro to CS I

Decomposition/Begin Functions

Odds and Ends...

- Exam I – Friday, 10/20
- Keep working on Assignment #3!!!
 - Design due Sunday on Canvas!!!
- Don't miss Demo, and be patient waiting
- READ, READ, READ!!!
- Ask TA pointed questions
- Just THINK! KISS 😊

Finish cin...

- What happens when we remove the `cin >> s;` before the `getline(cin, s);`?

Programming Errors

- Syntax errors
 - Misuse of C++ language
 - How are they caught? *compiler*
- Logic errors
 - Doesn't perform task correctly (aka. bugs)
 - How are they caught? *you!*
- Runtime errors
 - Stops your program from running
 - How are they caught? *you!*

Syntax Error Examples

- Missing main function
- Use of identifier not declared
- Misspelled Words
- Forget a Semicolon
- Forget Required Keyword
- Missing quote, curly brace, and parenthesis
- Use of single quotes instead of double

Logic Error Examples

- Poorly written programs
 - Add instead of subtract (incorrect operation)
 - Using last two digits for date
 - Same error message for different errors
 - Program that never ends
 - Add one to the largest integer (could be syntax)

Runtime Error Examples

- Segmentation fault or Core dump
 - Read a file that doesn't exist
 - Go outside of memory bounds
 - Infinite loop that eats memory
 - Divide by variable that is zero

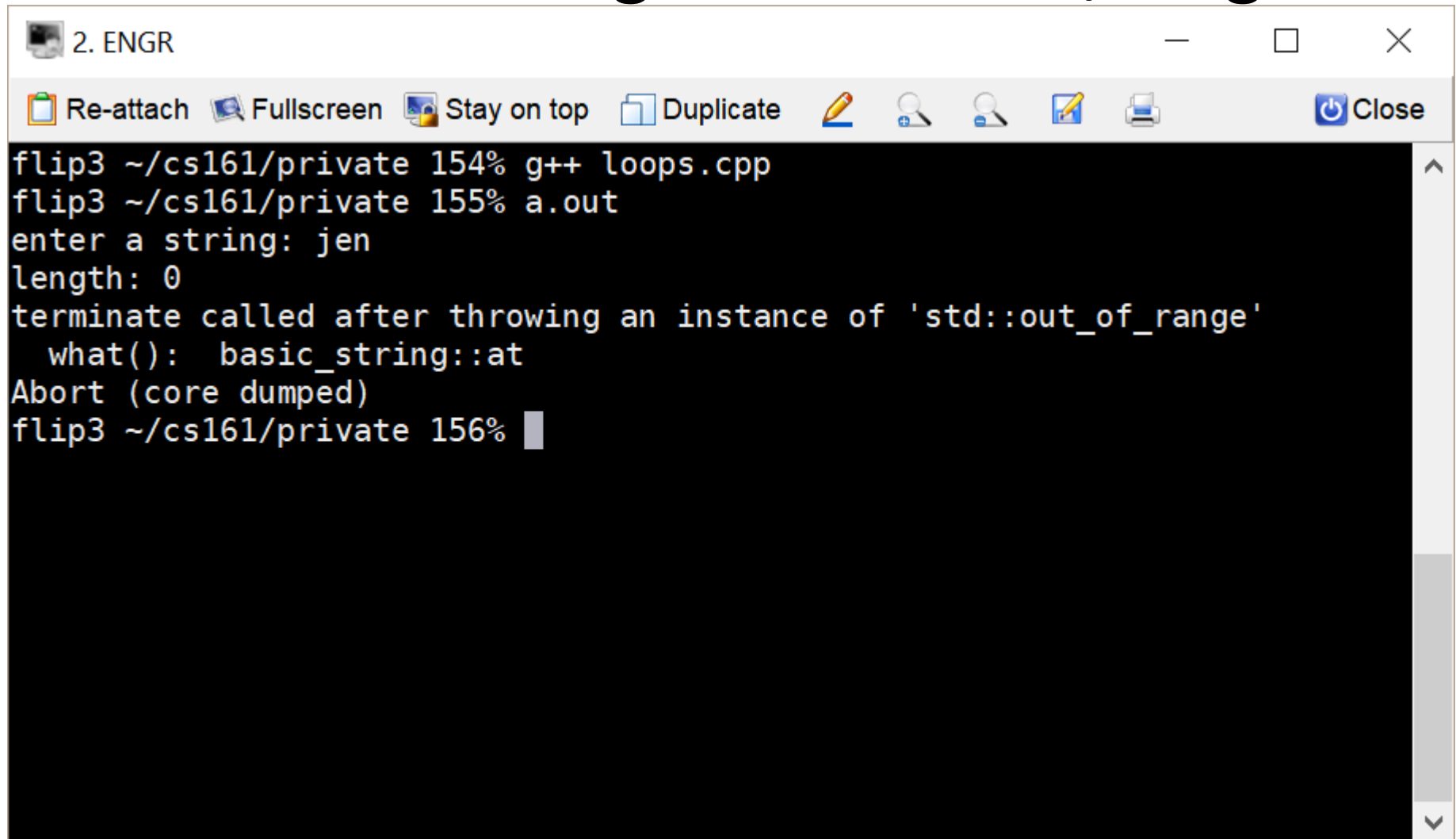
Debugging Errors

- Syntax:
 - **READ compiler errors** (pay attention to line #)
 - Use **google** to search for error
- Logic/Runtime
 - Use **std::cout** to find where the code is breaking
 - Print variable values
 - Print indicator messages
 - **Trace** through the code
 - **Comment** out code

above
below


```
1 #include <iostream>
2 #include <string> //c++ strings
3 #include <cstdlib> //atoi()
4
5 using namespace std;
6
7 int main() {
8     int x;
9     float f;
10    string s; //create a string object
11    bool bad; //create a flag to indicate bad or good data
12
13    //cin and getline difference: cin stops reading at any whitespace
14    //and stays at the whitespace where it left off reading, getline
15    //reads until a newline is encountered and moves past the newline
16    cout << "enter a string: ";
17    //Having both cin and getline will create a runtime error with string
18    //input not including any spaces and ends with newline because the
19    //cin stops at \n, but getline reads until seeing \n, which is where
20    //it left off from the cin. This means getline reads nothing and we
21    //can access the first character if there isn't one there!!!
22    cin >> s; //read a string of chars from the user until whitespace
23    getline(cin, s); //read chars from the user until a newline
24    cout << "length: " << s.length() << endl;
25    cout << "First char: " << s.at(0) << endl; //0 is 1st location
26    cout << "The whole string: " << s << endl;
27
```

Runtime error caused by s.at(0) when there is a string with 0 chars/length



A terminal window titled "2. ENGR" showing the execution of a C++ program. The user enters "jen" for a string, but the program incorrectly reports the length as 0. This causes a runtime error: "terminate called after throwing an instance of 'std::out_of_range'", followed by "what(): basic_string::at" and "Abort (core dumped)".

```
flip3 ~/cs161/private 154% g++ loops.cpp
flip3 ~/cs161/private 155% a.out
enter a string: jen
length: 0
terminate called after throwing an instance of 'std::out_of_range'
what(): basic_string::at
Abort (core dumped)
flip3 ~/cs161/private 156% █
```

Decomposition

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

Functions

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

Predefined Functions

- sqrt() ?
- pow()
- abs()
- rand() ← had info returned
- srand() ← put something here / argument
- What is the difference b/w srand() and others?

Hint: may be useful in Assign #3

no return value

Procedural Decomposition

- Functions

- `int main() { }`
 - User defined

- `void draw_box() { }`

- Function Call

- `draw_box();`

definition

what return

name of function

what is part of function

no type

no type

Procedural Decomposition

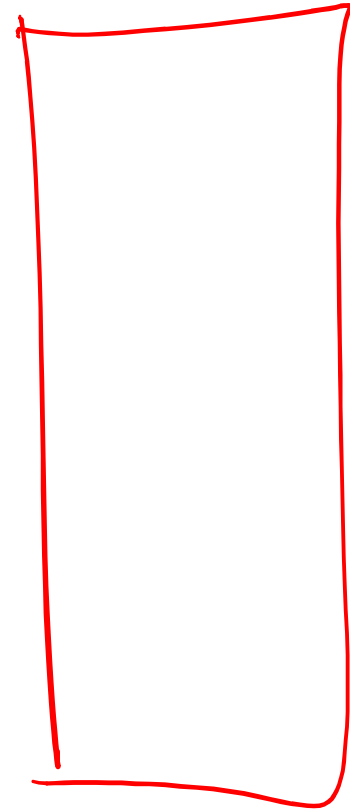
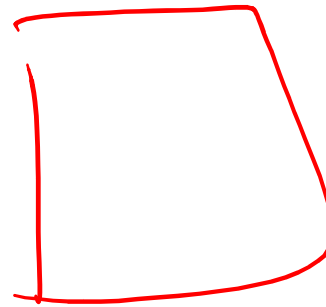
```
#include <iostream>
using std::cout;
int main() {
    cout << "+-----+\n";
    cout << "|         |\n";
    cout << "+-----+\n";
    cout << "+-----+\n";
    cout << "|         |\n";
    cout << "+-----+\n";
    return 0;
}
```

```
#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";
}
```

prototype

Functions Calling Other Functions

```
#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";
}
```



Generalization

/ Abstraction

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