

Lab 1 Debrief

Rollen D'Souza
rs2dsouz@uwaterloo.ca

University of Waterloo

06 October 2016

Outline

1 General Feedback

- Source Control
- C Programming

2 Lab Specifics

- Lab Solution
- Advice for the Future

Using Subversion

- Setup your SVN **before** creating a project!

Using Subversion

- Setup your SVN **before** creating a project!
- Commit code that compiles.

Using Subversion

- Setup your SVN **before** creating a project!
- Commit code that compiles.
- Commits should add/remove features or fix bugs, in general.

Using Subversion

- Setup your SVN **before** creating a project!
- Commit code that compiles.
- Commits should add/remove features or fix bugs, in general.
- Please, only one copy of your assignment.

Why only one copy

Example

```
svn log . -v
```

```
r1138 | | 2016-09-29 11:29:26 -0400
```

```
r1137 | | 2016-09-29 11:28:58 -0400
```

```
r1136 | | 2016-09-29 11:28:33 -0400
```

Why only one copy

Example

```
svn log . -v
```

```
r1138 | | 2016-09-29 11:29:26 -0400
```

```
r1137 | | 2016-09-29 11:28:58 -0400
```

```
r1136 | | 2016-09-29 11:28:33 -0400
```

Example

```
svn update -r 1136
```

```
svn update -r {2016-10-06T11:30}
```


C Arrays and Strings

What is wrong with the following piece of code?

Example

```
1 char buffer[500];  
2 readIntoBuffer(buffer);  
3 printf(buffer);
```

C Arrays and Strings

What is wrong with the following piece of code?

Example

```
1 char buffer[500];  
2 readIntoBuffer(buffer);  
3 printf(buffer);
```

What if readIntoBuffer didn't write to buffer?

C Arrays and Strings

Always initialize your arrays!

Example

```
1 char buffer[500] = { 0 };  
2 readIntoBuffer(buffer);  
3 printf(buffer);
```

Switches

Example

```
1 switch(character)
2 {
3     case 'A':
4     case 'a':
5         // code here.
6     break;
7
8     case 'B':
9     case 'b':
10        // more code here.
11    break;
12 }
```



Lab Solution

See Code in Repository.

Solution Take-Aways

- Tempted to hard-code a number? Use a `#define` or `const int` instead.

Solution Take-Aways

- Tempted to hard-code a number? Use a `#define` or `const int` instead.
- Use more than one file.
 - `extern`
 - forward declarations

Solution Take-Aways

- Tempted to hard-code a number? Use a `#define` or `const int` instead.
- Use more than one file.
 - `extern`
 - forward declarations
- Functions are for reusable code.

Solution Take-Aways

- Tempted to hard-code a number? Use a `#define` or `const int` instead.
- Use more than one file.
 - `extern`
 - forward declarations
- Functions are for reusable code.
- Use the C Standard Library
See <http://en.cppreference.com/w/c>

Advice

- Be Careful and Methodical.

Advice

- Be Careful and Methodical.
- Ask for help early!
 - Piazza
 - Email
 - Your Peers

Advice

- Be Careful and Methodical.
- Ask for help early!
 - Piazza
 - Email
 - Your Peers
- If you are unsure how to make your code better, talk it out!