Announcement

- Wrap up Project 3
  - Please let me know how I can help you
  - Don’t work till you drop
  - You might have other classes, I understand

- Grades
  - Please contact the Tas if there are any grade anomalies
Outline

- Will cover a Review today
  - Our goal was to bridge theory and practice of programming with CS

- Expected thorough knowledge in
  - Perl
  - C
  - C++

- Basic knowledge in shell script and unix tools used throughout the semester

Exam

- Basic exam should be completed in 2 hours

- Mix of definitions and programming analysis questions

- Open notes and books
  - Please don’t bring the whole library
  - Read over the labs and class notes
Perl

- Perl Basic programming
  - Scalars
  - Arrays
  - Hashes
  - Subroutines
    - Should know how to write, and how things are passed in, and how to call them
  - Perl debugger
    - Basic commands, how to call, why we use it
  - File handles
    - Open, close, processing

Perl knowledge

- Should know how to read/write perl code
- How to create classes
- Pass by references
- Dereference a reference
- What's the whole business with bless
Perl include path

- When you include a perl module
  - use foo;
    - What happens ???

- Looks for foo.pm
- use EXAM::foo
  - Will look where ??
- Perl -V
@inc

- Search path for stuff
- Some system local directory included
- Which means can use packages by having them local
- Else
- use lib “some/path”;

More

- Focus on C
  - Language basics
  - Main ideas behind design
  - Memory manipulations
  - Custom types
- Focus on CPP
  - Strongly typed
  - Classes and OOD
  - Inheritance
- Shell programming material
- Very basics ruby and ruby rails
### Sample Definition List

- `$ENV`
- `foreach`
- `for`
- `while`
- `next`
- `last`
- `unless`
- `until`
- `use strict`
- `$/`
- `$_`
- `$$`
- `$’`
- `my`
- `local`
- `die`
- `!`
- `$0`
- `@`
- `print`
- `=~`
- `~`
- `<>`
- `&&=`
- `chop`
- `exists`
- `eof`
- `split`
- `chomp`
- `our`
- `fork/exec`
- `open`

---

### Regular Expressions

- **What they are**
- **How to use them in perl**
  - Matching
  - Substitution
    - Flags and modifiers
- **See the lab on this**
CGI

- What is CGI?
- How and why we use cgi
- Typical errors
- Getting and processing user input
- Html formatting basics
  - Tags, webpages, etc
- Security related to CGI

C

- What is gcc, how do we use it
- Language basics
  - Basic data types
  - Basic loops
  - printf/scanf formatting and usages
  - Casting
  - Bitwise operators
  - String and string functions
- Makefiles and linking
  - Object files
Sample programming question

- What is wrong with this script?

```perl
#!/usr/local/bin/perl
# print the longest line in a file, excluding newline character.
while (<>) {
    chomp;
    if (length ($_) > length ($maxlength))
    {
        $longest = $_;
    }
}
print "The longest line is $longest.
";
```

sample

1. Write a perl script to show the current working directory.

2. write a program in perl to open a file and print each line out backwards
C

```c
#include

void main()
{
    int z = 2;
    z = 7;
    for (i = 1; i < 5; i++) {
        printf("%d\n" i*z);
        z += 2
    }
    printf("\n");
}
```

This program contains 5 errors. Circle each of these 5 errors in the code above and by each circle, write a short sentence describing the error.

cgi

- Write a simple html page to replace google’s main page, and a perl script to grab the user’s search term and store it in a log file named searched.txt
Theory

- How do you set the file permissions so that everyone can read some file tempsig.dat but not write to it?
- During running gcc, you notice .o files being created, what are they?
- Can you think of a reason why strings in perl needs the equals function and not the == ?
- The CGI specifies an interface whereby a CGI script called with method POST should expect to receive the data from the server in which variable? How is the GET method different?

Word list

- Compiling
- Linking
- Reference parameter
- Variable scope
- Stdio.h
- Stdlib.h
- cout
- cast
- Inline
- Linked list
- Preprocessor
- Typedef
- Struct
- Pointer
- Void pointer
- . Vs ->
- Function pointer
- Reference
- const
- malloc
Word list II

- Huffman
- getopt
- constructor
- destructor
- iostream
- overloading
- extern
- private
- Public
- GDB
- Cgi
- GET/POST
- overload
- overriding
- Template
- This
- Friend class
- New/delete
- virtual

C

- Basic constructs
- Basic type
- Advanced types
- (review labs and class examples)
- Memory stuff – understand what is happening
- Arrays
- Functions
- Pointers
- Debuggers
C
- Working with CGI
- Working on different platforms
- Makefiles
- How we built libraries

C++
- Basic language
- Difference to c
- Classes
- Permissions
- new/free memory allocations
- Inheritance and polymorphism
- Keywords
- Working with files....
Sample exam

- You’ve done most of the work for the course, the exam is just to make sure you remember the important concepts
- Will be posted online

- Couple Definitions
- 2 code checking question
- Shell code question
- C++ class manip question
- Small CGI question

Thinking question

- Say you are writing code which uses a random number generator....

- What is important to know about it ?
- How can your code be affected ?

- If you crash, how to reconstruct events, since based on random numbers ??
Closing Remarks

- If you like this.....just the beginning
- If you didn’t ..... You now know how complicated it is....never trust a program 😊
- Hope you had a fun semester..
- Extra office hours this week, please sign up for a demo spot
  - Will contact you about it

feedback

- If you have filled out the course evaluation on course works
- Thanks
- Else
- Do it today
Reschedule the midterm

- Any interest to reschedule to Sunday afternoon??

- Vote:

- Anything else specifically not covered??

- Dud anyone do the sample question I passed around??