

Midterm Exam Review

1 First, some Humor

- The following was my daily UNIX fortune when I first logged in...

Class Review Questions:

- (1) If Nerd on the planet Nutley starts out in his spaceship at 20 KPH, and his speed doubles every 3.2 seconds, how long will it be before he exceeds the speed of light? How long will it be before the Galactic Patrol picks up the pieces of his spaceship?
- (2) If Roger Rowdy wrecks his car every week, and each week he breaks twice as many bones as before, how long will it be before he breaks every bone in his body? How long will it be before they cut off his insurance? Where does he get a new car every week?
- (3) If Johnson drinks one beer the first hour (slow start), four beers the next hour, nine beers the next, etc., and stacks the cans in a pyramid, how soon will Johnson's pyramid be larger than King Tut's? When will it fall on him? Will he notice?

- Since we're reviewing for the midterm exam today, I thought this was hysterical

2 What the Midterm Covers

- From the syllabus, the midterm is 150 points, or 15% of your final grade
- The midterm will be one hour, from 12:30 to 1:30 pm on Wednesday, 9 July 2008
- Class *will continue* after the midterm at 1:45 pm and *attendance will be taken*
- The midterm will cover everything up to, and including, Monday's class (7 July 2008)
- Topics covered in Labs, Quizzes, and Homeworks will be included
- Quiz questions that most people missed might appear again
- *Now, the details...*

2.1 C++ Stuff

Terminology We didn't cover much... only two things. Please, please, pretty-please, know what they are.

Data Types Know *what* the data types `int`, `float`, `double`, `char`, `bool` *are*. Know what they are used to *represent*. Also, *know the limitations of each!* What are the common problems of, say, `float`, and *why does it have such limitations?*

Type Modifiers We covered one. You should know what it is and how it works.

Operators You should know *all* of the operators below, how many arguments they take, if they are left/right associative, and and their order of precedence.

Arithmetic	<code>+</code> , <code>-</code> , <code>*</code> , <code>/</code> , <code>%</code>
Boolean	<code> </code> , <code>&&</code> , <code>!</code>
Relational	<code>==</code> , <code>!=</code> , <code>></code> , <code>>=</code> , <code><</code> , <code><=</code>
Assignment	<code>=</code> , <code>+=</code> , <code>-=</code> , <code>*=</code> , <code>/=</code> , <code>%=</code> , <code>++</code> , <code>--</code>

Control Structures Know what `if`, `if/else`, and `switch` all accomplish and how they work.

Looping Structures Know what `while`, `do-while`, and `for` all accomplish, how they accomplish them, and what situation each is best suited for

- Remember, you will **not** have to write C++ code on the exam (or any exam/quiz)
- So how will this knowledge be tested?
- I'll just take your word that you have, in fact, learned the material

- ... *but we don't live in perfect-ville.*
- **Code Comprehension Questions** will probably make up half the exam.
- There will also be a question with various **Expressions** similar to what was on today's quiz, and you'll have to add parentheses, evaluate the expression, and know what *type* the resulting value is

2.2 Example Code Comprehension Questions

- **Easy** For inputs of -10, 0, 10, 50, and 100, what does the following program output?

```
int age;
cin >> age;
if (age > 65)
    cout << "Senior rate applies";
else if (age < 12)
    cout << "Child rate applies";
else
    cout << "Regular rate applies";
```

- **Medium** What does the following program output?

```
int d = 5;
while ( (d < 10) && (d >= 5) )
    d *= 5;

cout << d;
```

- **Difficult** What does the following program output?

```
int sum = 0;
for (int count1 = 1; count1 <= 10; count1++)
    for (int count2 = 0; count2 < count1; count2++)
        sum++;

cout << sum;
```

2.3 Software Design Stuff

- As stated the first day of class, the focus of this course is *really* on good software design principles and problem solving strategies.
- Before either of these can be studied heavily, we must first learn how to interact with a computer
- Thus, most of this class is actually “just” programming
- *But!* We *have* covered the principles of *good software design* and how to *use them*

- So, for the midterm, you should...
 1. Know what they are
 2. *Know how to apply them*
- There will be a question on the midterm asking you to develop an algorithm to solve a problem
- You will **not** have to write C++ code, but you will have to develop an algorithm (just like in the “Design Example” lecture)