CSE 232
First Midterm, Overview:

1. Getting Started
   1. we got started

2. First Program Stuff
   1. Compiler vs. Interpreter
      a. do you know all the steps to create an executable?
   2. Variables are declared
      a. indicate the type of the variable
      b. much of C++ (functions, STL) depend on types being available at compile time
         i. compile-time vs. run-time?
   3. Symbols are overloaded to
      a. required context to figure it out
   4. Basic program stuff
      a. what is an include file?
      b. what is a namespace
         i. does using namespace std do that I don't want?
         ii. how do I want you to do it?
         iii. what operator is ::
      c. two types of comments
      d. what is a block? How is it indicated?
      e. what purpose does whitespace/indentation serve in C++
      f. cout and cin, insertion op and extraction op.
         i. what does endl do (two things)
   5. what does it mean to have a function overloaded?

3. Types
   We spend an awful lot of time on types
   1. Do you know the basic, inbuilt types? What are their differences? Can they vary?
      What should you use when in doubt?
   2. Ways to do initialization
      a. don't do any (what do you get?)
      b. assign, paren, block
         i. what are the differences?
   Type modifiers. Things we can add on top of the type
   3. unsigned, for what kind of types?
   4. What does the compiler track with respect to variables?
   5. References
      a. how indicated?
      b. what does a reference mean (w.r.t. the stuff the compiler tracks)
      c. is it a new object? does it have to be initialized (why or why not)?
      d. what types are var1 and var2: int & var1, var2;
6. Pointers
   a. how indicated?
   b. what value does a pointer hold?
   c. is it a new object? Does it have to be initialized (why or why not)?
   d. what is the size of a pointer type (what question do you have to ask first)?

7. & and * in an expression
   a. what do these represent? Can you use them?
   b. what does the following print: int *ptr; cout << *ptr;
   c. what does the following print:
      i. int x=10; int &r_x=x; int *p_x=&r_x; *p_x=5;
      cout<< x;

8. Constants
   a. how indicated?
   b. what does it do to a value?
      i. can I assign it?
      ii. can I copy it?
   c. does it need to be initialized?
   d. can I remove it from a value?
   e. can I add it to a variable making reference to a non-constant value
      i. example???
   f. What two things can be constant in a pointer?

9. C++11 stuff
   a. what is a typedef?
   b. what does auto mean
      i. what are the rules here?

4. Expressions
1. cout formatting
   a. know the various ways to set things
      i. what is special about setw?

2. cin formatting
   a. what does cin take as a default separator?
   b. what does noskipws do?

3. Numeric ops (you basically know these)
   a. know what int on int division yields
   b. hex and oct, how indicated?

4. What is the way to do a cast in C++? Can you write one?

5. Assignment, it returns a value. What problems does that cause? Can you chain assignment? Why or why not?

   a. how indicated
   b. What difference does it make? Be very specific?

7. Compound Assignments, a shortcut: Any operator can precede an equal sign. It has the following meaning:
a. \( a \text{ op } b \) means \( a = a \text{ op } b \)
b. \( a += b \) means \( a = a + b \)
c. \( a /= b \) means \( a = a/b \)

8. Booleans and Conditionals
   a. Truth and Falsity in C++
      i. False is represented by 'empty' things in each type: 0 (int), 0.0 (float), '' (string)
      ii. If it isn't false (see above), then it represents true

9. Relational Operators, know what they are!
   a. they return true or false

10. Chained/compound comparisons do not work like you want in C++:
    a. is the following legal? \( 0 \leq 15 \leq 10 \);
    b. if so what does it return?

11. Know the Logical Operators (&&, ||, !)
12. Short circuiting. Logical operators do not return just true and false. They return the first value in an expression which makes the value of the expression clear. This is called short circuiting
   a. \( 1 \text{ || } 2 \) returns 1
   b. \( 0 \text{ && } 27 \) returns 0

5. Control
1. Selection
   a. do one statement (how to change)
   b. how do you know what else goes with what if
   if (Boolean)
      statement;
   if (Boolean)
      statement;
   else
      statement;
   if(Boolean)
      statement;
   else if(Boolean)
      statement;
   else

2. Repetition
   a. While loop (top tested)
   b. for loop
      i. do you know the three parts?
      ii. how are they separated?
      iii. which are optional?
      iv. can you write an equivalent while for a for loop?
      v. what is the scope of a variable declared in a for loop?
3. what is the switch statement good for
   a. under what conditions would you use it?
   b. with what kind of data?

4. ternary operator
   a. can you write one
   b. what's special about it (compared to an if statement)

6. Functions
   1. Functions are an encapsulation of a program. They are useful because they support:
      a. reusable code (can be used in many places)
      b. encapsulated code (details of implementation are hidden)
      c. portable code as modules/libraries (can be imported)
   2. More on why functions?
      a. break a larger program down into smaller, understandable parts.
      b. for easier update or "refactoring". Refactoring, as applied to functions, takes a larger piece of code and breaks it down into smaller function pieces. This makes it easier to maintain.
   3. How to write a function
      a. should do "one" thing. It represents one "idea" to be implemented
      b. should not be long
      c. should be generic, that is it should be reusable (used in other code).
      d. Should be **readable**!!!
   4. general format:
      a. name
      b. return type (before the name), value returned must match that type.
      c. the params, each with a type.
         i. run-time error?
   5. Scope
      a. block defines a scope
      b. parameters of a function are in the function scope, as is any variable declared in the function
      c. Can have multiple scopes with variables of the same name. The local scope defines the value
   6. What happens during the pass of argument to parameter?
      a. unless we say otherwise, it is a copy?

6. Chars and Strings
   1. How do you indicate a character? What is it the type name?
   2. What is Unicode? What is it important?
   3. Do you know the basic char functions (isalnum, isalpha, isdigit, islower, etc.)
      a. good for a cheat sheet
   4. Strings are an STL class, what does that mean? Are they a base type (like int)?
      a. what can you do with an STL object you cannot with a base type?
5. Strings store a sequence of what?

6. Input of strings:
   a. what's the difference between a cin>>a_str and getline(cin, a_str)?

7. What are two ways to index the characters of a string? What is the difference?

8. Any difference between .length() and .size() methods?

9. What does string::npos represent (how do we use it in code)?
   a. what are those :: things again?

10. What is a size_type for a container? How does it differ from an int?

11. I love range based for loops. Can you write one?
    a. what does that auto mean in a range based for?
    b. what does a range for print during each iteration on a string? Why?

12. Couple ways to construct an STL object (including a string). Know what they mean.
    a. string s;
    b. string s(10, '=');
    c. string s{'a','b','c'};
    d. string s1(s2);

13. some string methods:
    a. substr
    b. push_back
    c. find (know how this one works, how do you know you found something).
       i. variations here. Do you know them (cheat sheet).