Chapter 1

1. What is software?
2. What are the software ideals?
3. What is the main goal for the software ideals?
4. What are the 4 stages for the process of developing a program?

Chapter 2

1. What is a programming language?
2. What is a statement?
3. What is a compiler?
4. What is the process of taking source code and producing an executable?
5. What is a linker?
6. What are the 4 types of errors?
7. What errors are easier to find?

Chapter 3

1. What is an object?
2. What is a type?
3. What is a variable?
4. How does extraction work or. what makes extraction stop?
5. What is an assignment?
6. What is an initialization?
7. What makes a valid name in C++?

Chapter 4

1. What is abstraction?
2. What is divide and conquer?
3. How do we use abstraction and divide and conquer to write programs?
4. What is an expression?
5. What are the two selection statements in C++?
6. What are the 3 main iteration statements in C++?
7. What is a pre-test loop?
8. What is a post-test loop?
9. How do pre and post test loops differ and how are they the same?
10. What is a function?
11. Why do we “like” functions?
12. How do you declare a function?

Chapter 5

1. What are the 4 types of errors?
2. How do you find a syntax error?
3. How do you find a linking error?
4. How do you find a run-time error?
5. How do you find a logic error?
6. How is a logic error different then a run-time error?
7. What is an exception?
8. What is a debugger?

Chapter 8

1. What is a definition?
2. What is a declaration?
3. What is scope?
4. What is lifetime?
5. What is global scope?
6. What is local scope?
7. Why are constants ok to be declared in the global scope but not variables?
8. What is pass-by-value?
9. What is pass-by-reference?
10. When should you use pass-by-reference? -or- What two questions need a yes to use pass-by-reference?

Chapter 9

1. What is a struct?
2. How is a struct different than a class?
3. What is a class?
4. What is private access?
5. What is public access?
6. Where, which access modifier, should you use for fields?
7. Where, which access modifier, is mainly used for methods?
8. What pointer points to the object that is running the method?
9. What is a constructor? What is it supposed to do?
10. How can you tell a method is a constructor?
11. What is an accessor (getter) ?
12. What is a mutator (setter) ?

Chapter 10

1. What is a stream?
2. How do you open a file for reading?
3. How do you open a file for writing?
4. What causes an input stream to fail?

Chapter 11

1. How do you control the number of digits displayed on a double?
2. How do you control how many characters are displayed on the next output?

Chapter 17

1. What is an array?
2. Where do arrays begin counting their indices?
3. What are two limitations of arrays?
4. What is a pointer?
5. How do you typically give a pointer it's address?
6. How do you release the memory that a pointer is pointing to?
7. Where do pointers typically get their memory?

Chapter 18

1. How is a pointer like an array? How is a pointer different than an array?

Scope Example:
For the next questions use this small program to answer the questions. Assume all #include and using directives as needed:

```
int x = 10;
int otherFunction()
{
        int z = x * 4; //line 10
        return z; //line 20
}
int function(int x)
{
        x = x * 2; //line 30
        int y = x + x + otherFunction(); //line 40
        return y; //line 50
}
int main()
{
        x = 0; //line 60
        int x = 7; //line 70
        cout << function(x); //line 80
}
```

1. On line 10, what is the value for x on the right side of the equals sign?
A) 10
B) 7
C) 0
D) 14
E) 28
2. On line 30, what is the value for x on the right side of the equals sign?
A) 10
B) 7
C) 0
D) 14
E) 28
3. On line 40, what is the value for x on the right side of the equals sign?
A) 10
B) 7
C) 0
D) 14