Chapter 1

1. What is software?

A collection of programs running on some computer

2. What are the software ideals?

Correctness, reliability, well designed, and affordability and maintable.

3. What is the main goal for the software ideals? \*\*\*\*\*

It should be invisible.

4. What are the 4 stages for the process of developing a program?

1: Analysis

2: Design

3: Programming

4: Testing

Chapter 2

1. What is a programming language?

A precisely defined language with a specific grammar and a well defined vocab that we want performed.

2. What is a statement?

A statement is a command given to the computer that instructs the computer to take a specific action, such as display to the screen, or collect input. A computer program is made up of a series of statements.

3. What is a compiler?

A compiler is a computer program (or set of programs) that transforms source code written in a programming language (the source language) into another computer language (the target language, often having a binary form known as object code).

C++ source code 🡪 C++ compiler 🡪 Object Code

4. What is the process of taking source code and producing an executable?

C++ Source code (.cpp) 🡪 *Compiler* 🡪 Object File (.obj) + Object File from Std Lib 🡪 *Linker* 🡪 .Exe

5. What is a linker?

Programs consist of serveral parts. A linker takes one or more object files generated by a compiler and combines them into a single executable program (.exe)

6. What are the 4 types of errors?

7. What errors are easier to find?

Chapter 3

1. What is an object?

An object is a region of memory with a type that specifices what kind of information can be placed in it. A named object is called a variable.

2. What is a type?

A type is a classification identifying one of the various types of data. Such as int, bool, double.

3. What is a variable?

A variable is a ‘named’ object and has a specific type. A name would be the declaration for whatever an object is. (int age = 0) is an example where int is the type, age is the name, the zero is the value that is held in an object, and the whole thing would be a variable.

4. How does extraction work or. what makes extraction stop?

It asks for an input and assigns it to whatever variable that the input was assigned to push the value into. It stops when it runs into something

5. What is an assignment?

Giving a variable a new value.

6. What is an initialization?

Giving a variable its initial value.

7. What makes a valid name in C++?

**8. What are the 7 built-in types?**

Bool, char, int, float, double, short, long

Chapter 4

1. What is abstraction?

Ignores details for an amount of time. Hides details

2. What is divide and conquer?

Writing code with functions while using header file, main file, and implementation file. This allows the code to be changed in one spot and reflect a change in another so that a constant code update isn’t needed. Using functions will ‘divide’ the code allowing specific calculations or sections to be done separately so that it can be quickly fixed/changed.

3. How do we use abstraction and divide and conquer to write programs?

By using functions and dividing up the work, we can ‘ignore’ the other stuff while focusing on functions and finish those.

4. What is an expression?

An expression computes a value from a number of operands. Example would be (a=1; b=2; c=a\*b;)

5. What are the two selection statements in C++?

If and switch

6. What are the 3 main iteration statements in C++? \*\*\*\*\*\*\*\*\*\*\*\*

While, for, do while

7. What is a pre­test loop? \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

pre test loop does it before (for and while)

may never execute ever if false

8. What is a post­test loop?

guaranteed to run at least once

9. How do pre and post test loops differ and how are they the same? \*\*\*\*\*\*\*\*\*\*\*

One runs before a loop and the other after.

10. What is a function?

A function is a named sequence of statements. It can return a result.

11. Why do we “like” functions?

We can split up the work. It makes computations logically separate, it makes the program cleaner, it makes it possible to recall the function, and it allows for easier testing and fixing/debugging.

12. How do you declare a function?

A function declaration is the first line of the function that gives a variable, type, and all the information associated with it. The declaration can be at the top of the implementation code, in the code, or defined in the header with a #include. You then call the function after its been declared in your main code.

Example: void \_\_\_\_\_\_\_(…);

Chapter 5

1. What are the 4 types of errors?

Compile-time errors(syntax and type errors), link-time errors, run-time errors, logic errors.

2. How do you find a syntax error? \*\*\*\*\*\*\*\*\*\*\*

Through your compiler

3. How do you find a linking error? \*\*\*\*\*\*\*\*\*

4. How do you find a run­time error? \*\*\*\*\*\*\*\*\*\*\*\*\*\*

the compiler finds it

5. How do you find a logic error? \*\*\*\*\*\*\*\*\*\*

6. How is a logic error different then a run­time error?

7. What is an exception?

A way to deal with an error.

8. What is a debugger? StChapter 8  
 a computer program that assists in the detection and correction of errors in computer programs.

1. What is a definition? \*\*\*\*\*\*\*\*\*\*

A definition is when the function is called in the main code and includes ‘defining’ what the function is or does. It’s what the code does, and pretty much all the function code.

2. What is a declaration? \*\*\*\*\*\*\*\*

Making a variable and ‘declaring’ it to be something

3. What is scope?

Scope is where a variable is visible and ‘accessed’ in the code until it ends and becomes invisible from that block.

4. What is lifetime?

Lifetime is the time duration where an object/variable is in a valid state for use.

5. What is global scope? \*\*\*\*

A global variable is declared at the top of the implementation code or at the beginning of the int main(). An example would be globally declaring (string age = old;). The string age would be known to be ‘old’ throughout the entirety of the code, even in the functions as this is a global variable. If I declare that locally in a function, then the rest of the code won’t know that the age = old.

6. What is local scope? \*\*\*\*\*

Something defined ‘right there’ in the function part that can’t rename something in the global function

7. Why are constants ok to be declared in the global scope but not variables?

constants can be global because they can't change. Global constants are nice because they can't be changed. Variables can lead to confusing code. Global bypasses security and that it is then bad.

8. What is pass­by­value?

Pass-by-Copy=Pass-by-Value. If I print out the page and give you the printout, I'm **passing by value**. Your page is a disconnected copy of the original. You won't see any subsequent changes, and any changes that you make (e.g. scribbling on your printout) will not show up on the original page. If you destroy the printout, you have actually destroyed **your copy** of the object - but the original web page remains intact.” –Stack Overflow

9. What is pass­by­reference?

“If I tell you the URL, I'm **passing by reference.** You can use that URL to see the **same web page** I can see. If that page is changed, we both see the changes. If you delete the URL, all you're doing is destroying your reference to that page - you're not deleting the actual page itself.

10. When should you use pass­by­reference? ­or­ What two questions need a yes to use

pass­by­reference?

If a function will modify a variable. Then you use pass-by-refernece with a & symbol.

It should also ask the questions of:

1. Does the function modify the parameter?

2. Does the calling need to know about the change?

Chapter 9

1. What is a struct? \*\*\*\*\*\*\*\*\*

Complex data type declaration that defines a physically grouped list of variables to be placed

under one name in a block of memory. A struct is a class where members are public by default

2. How is a struct different than a class?

a struct is public by default, a class is private by default

3. What is a class?

A user defined type. It is composed of built in types or other user defined types and functions

4. What is private access?

Something that only the function can privately access

5. What is public access?

Something public

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? \*\*\*\*

Constructors have no return types and have the same name as the main class

10. How can you tell a method is a constructor?

11. What is an accessor (getter) ? \*\*\*\*\*\*\*\*\*\*

Ways to access information out of a class

12. What is a mutator (setter) ?

Way to change the private data

**Chapter 10**

1. What is a stream?

How can you tell a method is a constructor?

2. How do you open a file for reading?

ifstream.open

ofstream

3. How do you open a file for writing? \*\*\*\*\*\*

4. What causes an input stream to fail?

If it tries to read data of a different type than what it tries to read. Fails if it runs out of input.

Chapter 11

1. How do you control the number of digits displayed on a double?

setprecision

2. How do you control how many characters are displayed on the next output?

setw

Chapter 17

1. What is an array?

A series of objects all of which are the same size and type

2. Where do arrays begin counting their indices?

Starts at 0

3. What are two limitations of arrays?

Fixed size and can only be one type

4. What is a pointer?

variable that holds an address

5. How do you typically give a pointer it’s address?

You use the word 'new'

6. How do you release the memory that a pointer is pointing to?

Set it to null.

7. Where do pointers typically get their memory?

They get it from the free-store. The heap, or dynamically allocated place

Chapter 18

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

Pointer is a memory address. It's like an array in that it keeps an idea of is position

Pointer can point to an array. But you can dynamically allocate it

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