Study Guide 1

What is an example of a command line parameter being used? What is argc? How do you input an argument with spaces? What is argv? What is argv[0]? Why is it easier to put an argument in a string rather than use argv directly?

Slide 3,4

What happens if the inputted arguments do not follow the rules of the program (3 things)? How can using command line parameters get complicated?

Slide 5

What are the 3 functions to handling an exception? What happens when an exception is thrown (3 things)? What happens if a catch block is in the code? What happens when an exception is thrown in another function?

Slide 8,9

When there is a problem with an inputted argument, what is the temptation for the programmer to do? What should the programmer do? What is the functions job if there is an issue? What happens if an exception propagates past main? How do you use the catch block?

Slide 10,14,15

What is the function signature? What can lead to a compiler error with overloaded functions? What does the compiler look for when looking at overloaded functions? Why are function declarations always at the top of the function before main? What does #include essentially do?

Slide 18,19

What is function overloading? What are the different parameter configurations? Under what circumstances is it allowed to overload a function? Describe the process for determining whether or not to overload a function.

Slide 20,21

What are examples of common parameter values?

Slide 23,24,25,26,27,28

What does adding default parameters do to a function? What are the 4 notes about using default parameters? What is the new updated logic with using overloaded functions and default parameters?

Slide 29, 31,32

What are some different ways of organizing separate files? What is the typical structure for using separate files? What does this method prevent? What are the 3 methods for avoiding re-definitions? What are the 4 header guard functions and which one is superior but not commonly used?

Slide 35,36,37,38,39

What is the different between ‘ “” ‘ and “<>” when using #include?

Slide 42

What is a recursive function both mathematically and computer Science wise? What is an example of a recursive function?

Slide 49,50,51

Memorize the flowchart for a recursive function.

Slide 52,53

What are the parts of a recursive function? What happens if its not possible to get to the base case of a recursive function? What happens to the stack frame everytime a recursive function is called? What happens if the recursion becomes infinite (during runtime) and what is this called?

Slide 54

What are the 2 rules for writing recursive functions? Why do we use recursion? What is recursion a type of? What do you have to think about the stack frame whenever calling a recursive function?

Slide 55,56

What is the difference between recursion and loops and why is one sometimes preferred over the other?

Slide 57

What is debugging and how do we do it (without using IDE tools)? What are the 2 methods to simplifying debugging? What do you do when using print statements to debug code? What is a debugger and how is it used? What are breakpoints and how do they help with debugging? What are the 3 functions that a debugger uses and what are the differences between them?

Slide 61,62,63,64