Daniel Avila

CPSC 121

Chapter 6 Questions || Page 395

#1-10, 13-20, 23, 24, 26, 27, 28, 34, 36, 39(a, b, d)

10-15 Written Questions || 3-4 Written Questions - \*1 on 2D Arrays\*

1. The header is the part of a function definition that shows the function name, return type, and parameter list.
2. If a function doesn’t return a value, the word void will appear as its return type.
3. If function showValue has the following header: void showValue(int quality) you would use the statement showValue(5) to call it with the argument 5.
4. Either a function’s prototype or its definition must precede all calls to the function.
5. Values that are sent into a function are called arguments.
6. Special variables that hold copies of function arguments are called parameters.
7. When only a copy of an argument is passed to a function, it is said to be passed by value.
8. A(n) prototype eliminates the need to place a function definition before all calls to the function.
9. A(n) local variable is defined inside a function and is not accessible outside the function.
10. Constant Global variables are defined outside all functions and are accessible to any functions in a program.
11. If a function has a local variable with the same name as a global variable, only the local variable can be seen by the function.
12. Static local variables retain their value between function calls.
13. The return statement causes a function to end immediately.
14. Default arguments are passed to parameters automatically if no argument is provided in the function call.
15. When a function uses a mixture of parameters with and without default arguments, the parameters with default arguments must be defined last.
16. The value of a default argument must be a(n) literal.
17. When used as parameters, referenced variables allow a function to access the parameter’s original argument.
18. Reference variables are defined like regular variables, except there is a(n) ampersand in front of the name.
19. Two or more functions may have the same name, as long as their parameter lists are different.
20. What is the advantage of breaking your application’s code into several small functions?

Ability to debug faster and more efficiently.

Helps coder to be time efficient know what will work with this if not that.

More direct fixed rather than searching for problem.

1. When a function accepts multiple arguments, does it matter what order the arguments are passed in?

Yes, because if not, then you could be able to pass an integer value with a string, a char with a bool and so on, so in order to compile correctly and have the right data values, they need to be the same.

1. What does it mean to overload a function?

To manually input the arguments that the function would ask for and or have set values for fluctuating variables.

1. If you are writing a function that accepts an argument and you want to make sure the function cannot change the value of the argument, what should you do? Make it a constant value so that value doesn’t change with the code compiling
2. A program contains the following function.

int cube(int num)

{

return num \* num \* num;

}

Write a statement that passes the value 4 to this function and assigns its return value to the variable result.

Int result;

Result = cube(4);

1. A program contains the following function.

void display(int arg1, double arg2, char arg3)

{

cout << “Here are the values: “

<< arg1 << “ “ << arg2 << “ “ << arg3 << endl;

}

Write a statement that calls the function and passes the following variables to it:

int age;

double income;

char initial;

display(age,income,initial);

1. Each of the following functions has errors. Locate as many errors as you can.
2. void total( int value1, value2, value3) //arguments 2 and 3 are not initialized with a data type

{

return value1 + value2 + value3;

}

B)double average(int value1, int value2, int value3)

{

double average; //has the same name as the function header

average = value1 + value2 + value3 / 3; // does not match the data type that it wants to return, integers will

//never become double values

}

D)void getValue(int value&)// ampersand goes before the variable name

{

cout << “Enter a value: “;

cin >> value&; // Do not need to insert the ampersand when receiving the user input

}