**Computer Science 1**

**Final Exam Vocabulary List**

1. **Bit** –smallest unit of information processed by a computer (0 or 1)
2. **Byte** – eight bits
3. **Source Code** –aprogram written in a computer language
4. **Header File** – an existing C++ file that provides data and function declarations to clients
5. **High-level Language** – a language that uses instructions that resemble human languages and can be run on all computers regardless of manufacturer
6. **Comments** – non-executable statements used to make a program more readable
7. **Primitive Data Types** – also known as built-in data types: *int, double, char, bool*
8. **Literal value (Literal constant)** – an element that evaluates to itself and is not contained within a variable

Ex) cout << “house” << endl; // “house” is a literal constant

1. **Floating Point Number** – A number that includes a decimal value OR a number written in scientific notation to account for very large or very small values
2. **Operand** – a quantity or variable on which the operation is to be done
3. **Binary Operator** – an operator that operates on two operands
4. **Unary Operator** – an operator that operates on a single operand
5. **Precedence** – priority in importance, order, or rank
6. **Assignment Statement** – stores a new value into a variable
7. **Syntax** – the formal rules governing construction of valid statements in a programming language
8. **Cast** – converting a value of one type to a value of a different type
9. **Syntax Error** – an error that results from the violation of a programming language’s syntax
10. **Run-time error** – An error detected during the execution of the program

Ex)Attempting to divide by 0 at some point in a program

1. **Logical Error** – An error such that the program runs but unexpected results are produced
2. **Bug** – a programming error
3. **Debugging** - the process of eliminating “bugs” or errors from a program
4. **Scope** – where a variable can be used within a program
5. **Index** – an element’s specific location in an array
6. **Element** – an item within an array
7. **Function** – a specific block of code that can be called by name and return, at most, 1 value
8. **Calling Function** – a function that summons (makes a call to) a different function
9. **Called Function** – a function that is summoned into action by its reference in another function (often the *main()* function)
10. **Parameter (argument)** – value that is passed into a function when the function is called
11. **Function Prototype** – a declaration statement for a function
12. **Function Overloading** – occurs when the same function name is used for more than one function. The compiler must be able to determine which function to use based upon the function parameters.