

Unit 3_Intro To Programming Test Review

Below are the primary Programming Terms you have been learning and applying in this unit over the past few weeks. The test will be made up of two components:

FIRST. A term from the list below will be presented and you will **DEFINE** it and **EXPLAIN** where and how you used it in this unit.

SECOND. I will ask you to define and compare related terms.

To assist you, I am providing you with the definitions for the terms below (Your Welcome!).

Note: You have been dealing with most of these concepts for years, even though you might not have known the concept's name. Many of these terms are just subgroups of a greater category, eg. A Conditional is a type of Abstraction, OR, a Loop is a type of Iteration. Also, you can apply many of these terms to an individual lesson that your completed, such as when you created an algorithm and then wrote a program to draw the *Crosses*, it was an example of Functions, Top Down, Abstraction, Efficiency, etc., etc.

Algorithm: A process developed to solve a problem, that is executed by a computer and is implemented using programming languages.

Code: A precise sequence of instructions.

Program: When you look at all the lines of code that have been pieced together using a programming language to solve the problem...that is the program.

Why is a Programming Language Necessary (Lego)? Human language can be ambiguous. Computer language is unambiguous.

Creativity: Changing an algorithm in order to solve a new problem.

Sequence: Sequencing is the application of each step of an algorithm in the order in which the statements are given. (Sequencing is so fundamental, most times we don't think of it!)

Selection: It allows us to compare two Conditions, eg: a "True/False" Condition to determine which of the two parts of an algorithm should be used.

Efficiency: Achieving some desired outcome while minimizing wasted effort or resources.

Abstraction: Breaking a large complex program down into smaller workable pieces.

Function: A piece of code that the user can write in order to solve a problem. Two steps required: 1. *Define it* and 2. *Call it*.

Top Down Design: Repeatedly dividing a system into simpler subsystems.

API: A collection of commands made available to a programmer

Parameter: When you make a change to a function which typically affects the behavior of that function.

Iteration or Loops: is the repetition of part of an Algorithm until a condition is met or for a specified number of times.