CT 100

Programming and Algorithms
Outline

- Programming
- Scratch
- Algorithms
- Problem Solving
Algorithm

• A finite sequence of instructions that can solve a problem in a finite amount of time.
Programming

• An implementation of an algorithm in some programming language
• In general programming includes more than implementation of algorithms but that will be the focus of our discussion
• Syntax of a programming language
• Semantics of a programming language
Scratch

- Programming language and environment created by MIT to teach introductory programming
- Programs are created by “gluing” together blocks
Algorithm/Programming Concepts

• Variables
  – Simple variables
  – Lists
• Events
• Statements
• Loops
  – Repeat until
• Selection
  – If else
Variable

• A name that holds or refers to a value. The same variable can hold different values at different times in a program but at any particular time the variable only holds one value.

• In most of the examples we will look at the variables we use will refer to numbers or lists of numbers
Event

• An external action that the program reacts to.

• For example pressing a key or clicking the mouse

• In our examples we will look at programs that react to pressing keys on the keyboard
Statements

- Instructions in the programs
- For example assigning a new value to a variable
Loops

• A kind of statement that indicates that a group of statements should be repeatedly executed until some condition is true

• The most common loops we will use in Scratch are repeat loops.
Selection

• Selection is a kind of statement that can indicate either to execute a group of statements if some condition is true or to execute one of two groups of statements based on a condition

• The most common selection statements we will use in Scratch are if statements and if-else statements
Selection Sort Example

```plaintext
when key pressed
set i to 1
repeat until i > 9
set j to i + 1
set smallLoc to i
repeat until j > 10
if item j of nums < item smallLoc of nums
set smallLoc to j
change j by 1
set temp to item i of nums
replace item i of nums with item smallLoc of nums
replace item smallLoc of nums with temp
change i by 1
```