Unit 3—Algorithms and Programming
Chapter 1: Programming Languages and Algorithms
Study Guide

Big Questions

- Why do we need algorithms?
- How is designing an algorithm to solve a problem different from other kinds of problem solving?
- How do you design a solution for a problem so that it is programmable?
- What does it mean to be a creative programmer?
- How do programmers collaborate?

Enduring Understandings

- 1.1 Creative development can be an essential process for creating computational artifacts.
- 1.2 Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem.
- 2.2 Multiple levels of abstraction are used to write programs or create other computational artifacts.
- 4.1 Algorithms are precise sequences of instructions for processes that can be executed by a computer and are implemented using programming languages.
- 5.1 Programs can be developed for creative expression, to satisfy personal curiosity, to create new knowledge, or to solve problems (to help people, organizations, or society).
- 5.2 People write programs to execute algorithms.
- 5.3 Programming is facilitated by appropriate abstractions.

1. Take time to go through Unit 3 Stages 1-9 and read the student lesson plans focusing on the important concepts presented in the Background and Vocabulary. You should take notes on definitions, ideas, etc. Be sure to watch every video provided in each unit. Also, be sure to review the reflections from throughout the unit.

2. Make sure you complete every coding puzzle in all units! Since this is a programming/coding unit, you will need to be able to read and write code and understand the following concepts/commands including the proper notation.

   a. Basic Turtle Commands
      i. moveForward(pixels)
      ii. moveTo(x,y)
      iii. turnLeft (angle)
      iv. penUp ( )
      v. dot(size)
      vi. arcLeft(angle, radius)
      vii. penRBG(r,g,b,a)

   b. Functions
      i. Calling a function—with and without a parameter(s)
      ii. Define a Function—with and without a parameter(s)

   c. Loops & For Loops
      i. for (var i = start; i < end; i++) { }

   d. Random Number Function
      i. randomNumber(min,max)

3. Be prepared to be tested on the following topics:

a) Lesson 1: The Need for Programming Languages

b) Lesson 2: The Need for Algorithms
   a. Algorithm
   b. High Level Programming Languages
   c. Low Level Programming Languages

c) Lesson 3: Creativity in Algorithms
   a. Algorithm
   b. Iterate
   c. Selection
   d. Sequencing
   e. Flow charting
   f. Pseudocode

d) Lesson 4: Using Simple Commands
   a. Turtle Programming

e) Lesson 5: Creating Functions
   a. Abstraction
   b. Function
   c. Parameter
   d. Argument
   e. Variable

f) Lesson 6: Top Down Design

g) Lesson 7: APIs and Functions with Parameters
   a. API
   b. IDE
   c. Documentation

h) Lesson 8: Creating Function with Parameters

i) Lesson 9: Looping and Random Numbers
   a. For Loop
   b. Loop
   c. Randomness

Programs/Coding Examples

1. Read the following program and draw the result (including the ending state of the turtle). Assume that the turtle starts facing up. Label parts/lengths of your picture to show understanding of the code.

```
drawDesign(25,10);
function drawDesign(size, width) {
          penDown();
         penWidth(width);
         moveForward(size);
         turnRight(90);
         moveForward(size);
         moveForward(size);
         turnRight(90);
         moveBackward(size);
         turnLeft(90);
          penUp();
        }
```

2. Read the following program and draw the result (including the ending state of the turtle). Assume that the turtle starts facing up. Label parts/lengths of your picture to show understanding of the code.

```
for(var i=0; i<4; i++){
        moveForward(100);
        turnLeft(90);
        }
```

3. The following function was created, but an error message occurred when trying to run the program. Find and correct the problem in the function definition. Then call the function so that a star of length 30 is created.

```
function drawStar () {
        for(var i=0; i<5; i++) {
                moveForward (size)
                turnRight (144)
        }
}
```

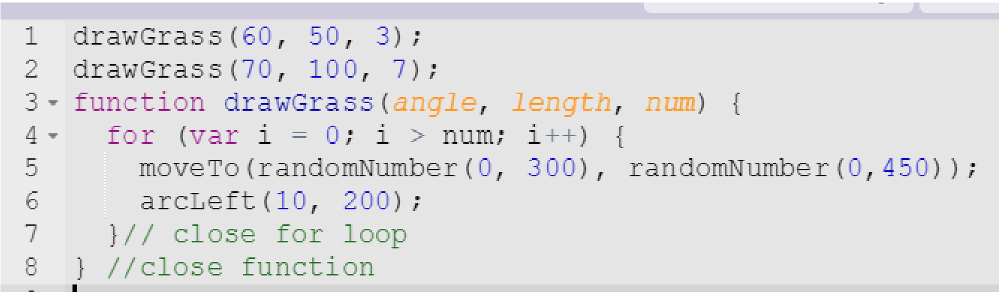
4. Read the following program and explain what it does.

```
        for (var i = 0; i < 50; i++) {
                penUp( );
                moveTo(randomNumber(0, 320), randomNumber(0, 450));
                drawSquare(randomNumber(25, 100));
        }
function drawSquare(size) {
        penDown();
        moveForward(size);
        turnRight(90);
        moveForward(size);
        turnRight(90);
        moveForward(size);
        turnRight(90);
        moveForward(size);
        turnRight(90);
        penUp();
        }
```

5. Explain how the program would be affected (if at all) if the drawSquare command was called outside of the “for loop.”

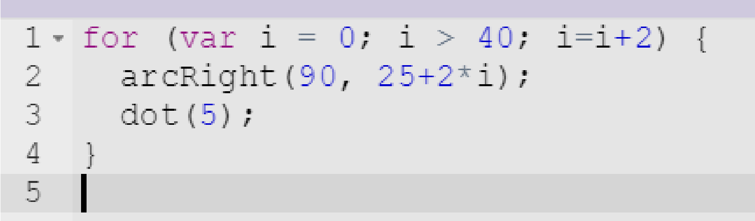
6. The drawSquare function includes some repetitive commands. Could a “loop” (or “for-loop”) be used within this function? Explain why or why not.

7. What is wrong with the following code?



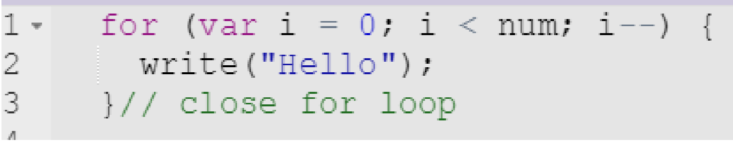
```
drawGrass(60, 50, 3);
drawGrass(70, 100, 7);
function drawGrass(angle, length, num) {
  for (var i = 0; i > num; i++) {
    moveTo(randomNumber(0, 300), randomNumber(0,450));
    arcLeft(10, 200);
  }// close for loop
} //close function
```

8. What is the output of this code?



```
for (var i = 0; i > 40; i=i+2) {
  arcRight(90, 25+2*i);
  dot(5);
}
```

9. What is wrong with the following code?



```
for (var i = 0; i < num; i--) {
  write("Hello");
}// close for loop
```

10. List the following in this code:

Parameter(s)

Argument(s)

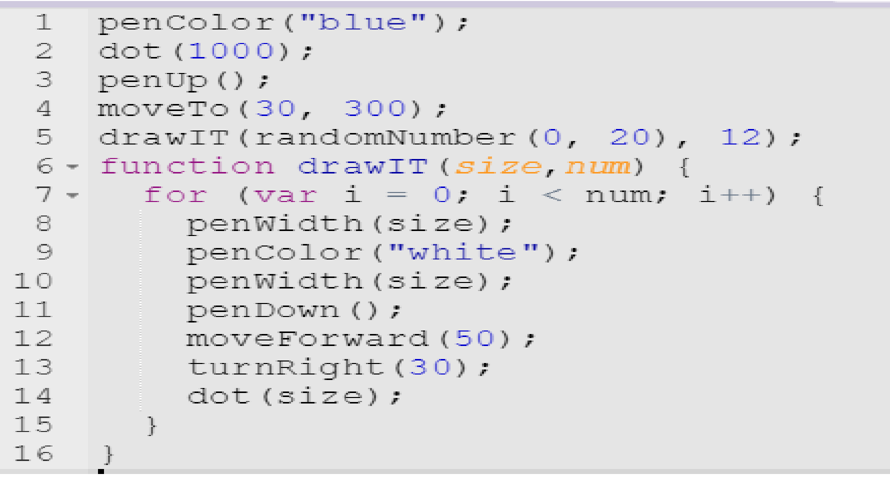
Control variable(s)

Control variable comparison

control variable change?

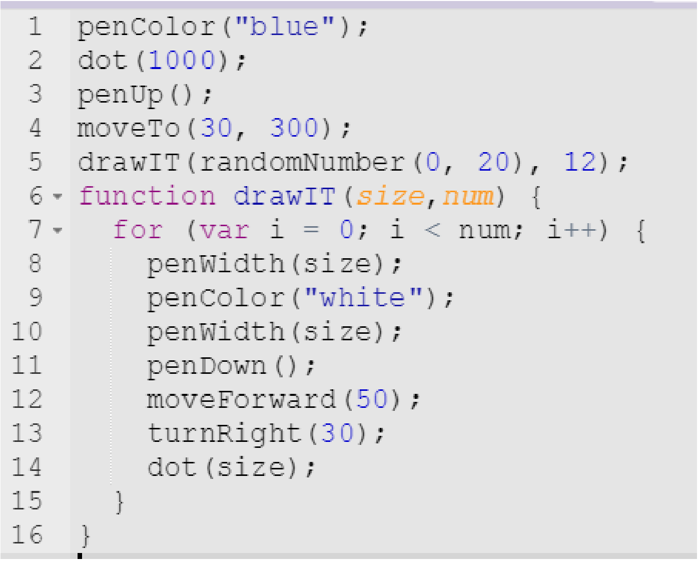
Function Definition

Function Call

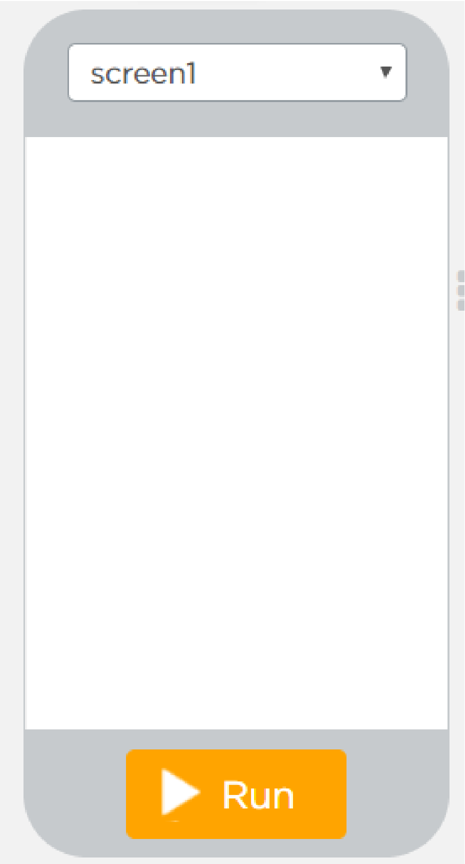


```
penColor("blue");
dot(1000);
penUp();
moveTo(30, 300);
drawIT(randomNumber(0, 20), 12);
function drawIT(size,num) {
  for (var i = 0; i < num; i++) {
    penWidth(size);
    penColor("white");
    penWidth(size);
    penDown();
    moveForward(50);
    turnRight(30);
    dot(size);
  }
}
```

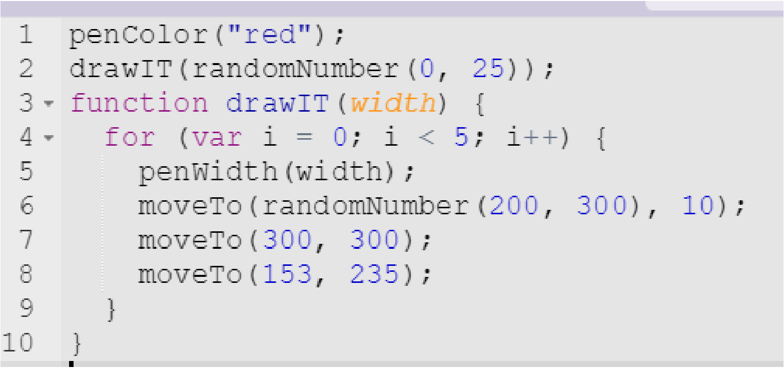
11. Draw the output on the screen provided. Be sure to be accurate in placement of output.



```
penColor("blue");
dot(1000);
penUp();
moveTo(30, 300);
drawIT(randomNumber(0, 20), 12);
function drawIT(size,num) {
  for (var i = 0; i < num; i++) {
    penWidth(size);
    penColor("white");
    penWidth(size);
    penDown();
    moveForward(50);
    turnRight(30);
    dot(size);
  }
}
```



12. Draw the output on the screen provided. Be sure to be accurate in placement of output.



```
penColor("red");
drawIT(randomNumber(0, 25));
function drawIT(width) {
  for (var i = 0; i < 5; i++) {
    penWidth(width);
    moveTo(randomNumber(200, 300), 10);
    moveTo(300, 300);
    moveTo(153, 235);
  }
}
```