Logic Chapter 3

William Elizondo

Short Answer:

1. How do modules help you reuse code in a program?

*Modules have a term called “code reuse” because you are writing code to perform a task once and then reusing it each time you need to perform the task.*

1. Name and describe the two parts that a module definition has in most languages.

*The header and the body. The header is the starting point of the module and the body is a list of statements that belong to the module.*

1. When a module is executing, what happens when the end of the module is reached?

*When the end of the module is reached, the computer jumps back to the part of the program that called the module, and the program resumes execution at that point.*

1. What is a local variable? What statements are able to access a local variable?

*A “local variable” is a variable that's declared inside a module and cannot be accessed by statements that are outside that module. Statements that are inside that module can only access that variable.*

1. In most languages, where does a local variable’s scope begin and end?

*It usually begins at the variable declaration and ends at the end of the module in which the variable is declared.*

1. What is the difference between passing an argument by value and passing it by reference.

*Passing an argument by value means that only a copy of the arguments value is passed into the parameter variable.  
When an argument is passed by a reference it means it’s passed into a special type of parameter known as a reference variable.*

1. Why do global variables make a program difficult to debug?

*Any statement in a program can change the value of a global variable.*

Algorithm Workbench

3. Look at the following pseudocode module header:

***Module myModule(Integer a, Integer b, Integer c****)*

Now look at the following call to myModule:

***Call myModule (3, 2, 1)***

When this call executes, what value will be stored in *a*? What value will be stored in *b?* What value will be stored in *c?*

So when it executes the values will be

a = 3

b = 2

c = 1

4. Assume that a pseudocode program contains the following module:

***Module display (Integer arg1, Real arg2, String arg3)***

***Display “Here are the values:”***

***Display arg1, “ ”, arg2, “ ”, arg3***

***End Module***

Assume that the same program has a main module with the following variable declarations:  
  
***Declare Integer age***

***Declare Real income***

***Declare String name***

Write a statement that calls the display module and passes these variables to it.

Answer:  
  
***Module main***

***Declare Integer age***

***Declare Integer income***

***Declare String name***

***Call display(age, income, name)***

***End Module***

6.What will the following pseudocode program display?

***Module main ()***

***Declare Integer x = 1***

***Declare Real y = 3.4***

***Display x, “ ”, y***

***Call changeUs (x, y)***

***Display x, “ “ , y***

***End Module***

***Module changeUs (Integer a, Real b)***

***Set a = 0***

***Set b = 0***

***Display a, “ “, b***

***End Module***

Answer:

*1, 3.4*

*0, 0*

7. What will the following pseudocode program display?

***Module main ()***

***Declare Integer x = 1***

***Declare Real y = 3.4***

***Display x, “ ”, y***

***Call changeUs (x, y)***

***Display x, “ “, y***

***End Module***

***Module changeUs (Integer Ref a, Real ref b)***

***Set a = 0***

***Set b = 0.0***

***Display a, “ “, b***

***End Module***

Answer:

1, 3.4

0, 0.0

0, 0.0

Debugging Exercise

1.Find the error in the following pseudocode.

***Module main ()  
 Declare Real mileage***

***Call getMileage()***

***Display “You’ve driven a total of ”, mileage, “ miles.”***

***End Module***

***Module getMileage()  
 Display “Enter your vehicle’s mileage.”***

***Input mileage***

***End Module***

Answer:   
The problem is with the variable “mileage”. It’s been declared in the main module but they’re trying to access it inside of a different module.

3.Find the potential error in the following pseudocode.

***Module main()***

***Call squareNumber(5)***

***End Module***

***Module squareNumber(Integer Ref number)***

***Set number = number^2***

***Display number***

***End Module***

Answer: In this code squareNumber module is defined to accept an “Integer Ref number” which means it expects a reference to an integer variable. The problem lies where when squareNumber is called in the main module it’s called with a literal value instead of a variable reference. The variable isn’t declared so it’s just a literal value and we can’t reference that.

4.Find the error in the following pseudocode.

***Module main()***

***Call raiseToPower(2, 1.5)***

***End Module***

***Module raiseToPower(Real value, Integer power)***

***Declare Real result***

***Set result = value^power***

***Display result***

***End Module***

Answer: The issue lies where having a real number multiply to the power of an integer. This can result in a type mismatch in data types or unexpected behavior.

Programming Exercises