LAB 4-1 ASSIGNMENT

Ch. 3 Modules

Start: 01/31/2024 6:11 PM
Name: Imani Hollie

LAB 4-1.1 – ALGORITHMS

Problem 1: Design a module named `timesTen`. The module should accept an `Integer` argument. When the module is called, it should display the product of its argument multiplied times 10.

```
Module timesTen(Integer a)
   Declare Integer y
   Set y = a * 10
   Display y
End Module
```

Problem 2: Examine the following pseudocode module header, and then write a statement that calls the module, passing 12 as an argument.

- `Module showValue(Integer quantity)`
- `Call showValue(12)`

Problem 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`?

The parameter list passes 3 to the `a` parameter, 2 to the `b` parameter, and 1 to the `c` parameter.

Problem 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.

- `Call display(age, income, name)`

Problem 5: Design a module named `getNumber`, which uses a reference parameter variable to accept an `Integer` argument. The module should prompt the user to enter a number and then store the input in the reference parameter variable.

```
Module getNumber(Integer Ref a)
   Display "Enter a number."
   Input y
```

```
End Module
```

Problem 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
  ```

The following pseudocode will output:

- ```
  1 3.4
  0 0
  ```

This is because the original values, `x = 1` and `y = 3.4`, were displayed, and then overwritten in the changeUs Module once it was called and displayed too.

Problem 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
  ```

The following pseudocode will output:

- ```
  1 3.4
  0 0.0
  ```

The last number, `b = 0.0`, is the only one to change as that's what it's set to in the changeUs Module.

LAB 4-1.2 – DEBUGGING EXERCISES

Exercise 1: Find the error in the following pseudocode:

```
1. Module main()
   2. Declare Real mileage
   3. Call getMileage()
   4. Display "You've driven a total of ", mileage, " miles."
5. End Module
6. Module getMileage()
   7. Display "Enter your vehicle's milage: "
   8. Input mileage
9. End Module
```

An error occurs on Lines 3 and 6: The `mileage` variable is declared inside the `main` module as a local variable, thus the `getMileage` module cannot access it. To fix it, pass it as a reference variable to the getMileage module on Lines 3 and 6 as follows:

```
1. Module main()
   2. Declare Real mileage
   3. Call getMileage(milage)
   4. Display "You've driven a total of ", mileage, " miles."
5. End Module
6. Module getMileage(Real Ref milage)
   7. Display "Enter your vehicle's milage: "
   8. Input mileage
9. End Module
```

Exercise 2: Find the error in the following pseudocode:

```
1. Module main()
   2. Call getCalories()
3. End Module
4. Module getCalories()
   5. Declare Real calories
   6. Display "How many calories are in the first food?"
   7. Input calories
   8. Declare Real calories
   9. Display "How many calories are in the second food?"
   10. Input calories
11. End Module
```

An error occurs on Lines 8 and 10: The `calories` variable has already been declared on Line 5. To fix it, rename the variable on Line 8 and 10 as follows:

```
1. Module main()
   2. Call getCalories()
3. End Module
4. Module getCalories()
   5. Declare Real calories
   6. Display "How many calories are in the first food?"
   7. Input calories
   8. Declare Real calories2
   9. Display "How many calories are in the second food?"
   10. Input calories2
```

```
11. End Module
```

Exercise 3: Find the potential error in the following pseudocode:

```
1. Module main()
   2. Call squareNumber(5)
3. End Module
4. Module squareNumber(Integer Ref number)
   5. Set number = number ^ 2
   6. Display number
7. End Module
```

An error occurs between Lines 4 and 5: The variable number is never declared and assigned the `Integer` data type. To fix it, you have to insert a Line as follows:

```
1. Module main()
   2. Call squareNumber(5)
3. End Module
4. Module squareNumber(Integer Ref number)
   ▪ Declare Integer number
   5. Set number = number ^ 2
   6. Display number
7. End Module
```

Exercise 4: Find the error in the following pseudocode:

```
1. Module main()
   2. Call raiseToPower(2, 1.5)
3. End Module
4. Module raiseToPower(Real value, Integer power)
   5. Declare Real result
   6. Set result = value ^ power
   7. Display result
8. End Module
```

An error occurs on Lines 4 and 5: The variables, specifically `Integer power`, have conflicting data types with `Real result` as the argument and the receiving parameter must be of the same data type. To fix it, you have to change the other variables to be `Integer` on Lines 4 and 5 as follows:

```
1. Module main()
   2. Call raiseToPower(2, 1.5)
3. End Module
4. Module raiseToPower(Integer value, Integer power)
   5. Declare Integer result
   6. Set result = value ^ power
   7. Display result
8. End Module
```