**CIS 129 Final Exam - Score:**

This final exam is to be done in pseudocode, not Java. In these pseudocode problems you will be asked to write or fix just a few coding statements. You do not have to write the entire program. When applicable, use these variable names: myInteger, myReal, myString, myBoolean, myIntegerArray. (100 points available)

**Declare Integer [ ] myArray = new Integer [50]**

**#1a – In pseudocode, write a call to a function that passes 1 integer variable and the integer array above (as a variable) and accepts a Boolean value in return. (2 points)**

**#1b – In pseudocode, write the function that accepts 1 integer parameter and an integer array and returns a Boolean. Search the integer array with a for-loop and if the integer parameter is found in the array, return false. If the integer parameter is not found, return true. (4 points)**

//Write your answer below:

Set myBoolean to myFunction(myInteger, myArray)

Module myFunction(integer, array)

For x=0 to x = arrayLength increment 1

If array[x] is equal to integer

Return false

Else

Return true

End Module

**#2a – In pseudocode, write a call to a module that passes 2 integer variables, 1 String constant, and 1 String literal as arguments. (2 points)**

**#2b – In pseudocode, write the module header that accepts 2 integer variables and 2 Strings as parameters. (2 points)**

//Write your answer below:

Call myFunction (myInteger1, myInteger2, MYCONSTRING, “myString”)

Module myFunction (integer1, integer2, string1, string2)

**#3 – This pseudocode has multiple problems. Fix the calling statement and the function that is supposed to accept 3 grades as parameters and return the average to a variable: (4 points)**

Call calcAverageGrades(Boolean grade5)

Module calcAverageGrades (grade1, Grade3, grade2)

Set averageGrade = grade + grade2 + grade3 / 3

End Module

//Write your answer below:

Call calcAverageGrades(grade1, grade2, grade3)

Module calcAverageGrades (grade1, grade2, grade3)

Set averageGrade = (grade1 + grade2 + grade3) / 3

End Module

**#4 – In pseudocode write a for-loop that decreases from 100 to 2 by 2s. Print the value of the loop variable in the loop unless the value is from 40 to 50 inclusive. Use only 1   
if-statement. (5 points) Do NOT write Java code!**

//Write your answer below:

For x=100 to x = 2 decrement -2

If x is in range 40 to 50

Continue

else

Display x

**#5 – What is the value of myAnswer after this code runs in each case below?   
(2 points each for a total of 10 points).**

switch (myAnswer){

case 5:

myAnswer = 15;

case 10:

myAnswer++;

if (myAnswer == 11)

myAnswer = 22;

break;

else

myAnswer--;

end if

case 15:

myAnswer = myAnswer + 10;

break;

case 20:

myAnswer = 5;

break;

case 25:

myAnswer = myAnswer + myAnswer;

default:

myAnswer++;

break;

**If… myAnswer = 10 then myAnswer = 22**

**myAnswer = 20 then myAnswer = 5**

**myAnswer = 5 then myAnswer = 15**

**myAnswer = 25 then myAnswer = 51**

**myAnswer = 17 then myAnswer = 18**

**#6 – What are the subscript values for this 2-dimensional array?   
Enter them in the array below… (4 points)**

|  |  |  |  |
| --- | --- | --- | --- |
| **0,0** | **0,1** | **0,2** | **0,3** |
| **1,0** | **1,1** | **1,2** | **1,3** |
| **2,0** | **2,1** | **2,2** | **2,3** |
| **3,0** | **3,1** | **3,2** | **3,3** |

**#7 – Using this property…**

**Private Real employeeSalary;**

**#7a - Write a getter. (2 points)**

Public Real getEmployeeSalary() {

Return this.employeeSalary;

}

**#7b - Write a setter. (2 points)**

Public void setEmployeeSalary (Real var) {

This.employeeSalary = var

}

**#8 – Debug this input validation code and fix it. There is lots wrong with it! (12 points)**

**Function String getNumberGreaterThanX (msg, x)**

**myInteger = getInteger(msg)**

**While myInteger < x**

**Display "The number must be < “, myInteger , “. Try again."**

**Input x**

**End While**

**Return x**

**End Function**

*(Keep the above code and rewrite the copy below)*

Function Integer getNumberGreaterThanX (msg, x)

myInteger = getInteger(msg)

While myInteger < x

Display "The number must be > “, x , “. Try again."

Input myInteger

End While

Return myInteger

End Function

**#9 – Write a standard one-function input validation routine as called below. The validation function should return an integer value greater than zero. (12 points)**

int goodValue = getValidNumberGreaterThanZero(message)

Function getValidNumberGreaterThanZero(message)

While goodValue is NOT > 0

Display “please enter a number greater than 0”

goodValue = input

End While

Return goodValue

End Function

**#10 – Do these statements evaluate to true or false? (2 points each; 20 total)**

NOT true evaluates to: false

true AND true evaluates to: true

false OR NOT false evaluates to: true

true OR false evaluates to: true

false AND false evaluates to: false

false OR false evaluates to: false

(NOT (true OR false)) AND true evaluates to: false

(NOT true) AND (NOT false) evaluates to: false

NOT (false OR (NOT true)) evaluates to: true

(NOT ((false AND false) AND (true OR false)) AND (true OR NOT false))

evaluates to: true

**#11a - Source code is… (2 points)**

a text listing of commands to be compiled or assembled into an executable computer program

**A file name example of source code relating to Java would be: (2 points)**

Sample.java

**#11b - Executable code is…** **(2 points)**

Code that causes the computer to perform indicated tasks according to encoded instructions

**A file name example of an executable file relating to Java would be: (2 points)**

Sample.exe

**#11c - The definitions of semantic and syntax errors are… (2 points)**

Semantic errors are when your code runs successfully but produces unexpected results while syntax errors are when the programmer writes code that does not correctly follow the rules of the programming language used.

**An example of a pseudocode semantic error would be: (explain why) (2 points)**

Call myFunction

myFunction

system exit

display myVariable

this is a semantic error because the code will run but will exit before anything is displayed.

**An example of a pseudocode syntax error would be: (explain why) (2 points)**

Call myFunction()

myFunction(

display (myVariable)

this will create a syntax error because the declaration of myFunction is missing the required )

**#12a – In OOP, encapsulation means… (2 points)**

binding together the data and functions that manipulate the data, and that keeps both safe from outside interference and misuse

**#12b – In OOP, a constructor is… and usually does this… (2 points)**

**A constructor is called to create an object. It prepares the new object for use and typically accepts arguments that the constructor uses to set variables**

**#12c –Complete this common phrase used in OOP: (2 points)**

**Private \_\_\_\_\_\_Good\_\_\_\_\_\_\_\_\_\_; Public \_\_\_\_\_\_\_Bad\_\_\_\_\_\_\_\_\_\_\_\_**