IT 1090C Computer Programming I
Prof. Tom Wulf / David Freeze

Lab 2 – Flow Charts and Pseudo Code Conditionals

Pts 10

Mini-lecture:

We use pseudo code and flow charts to capture the logic of a program prior to implementing it with code. In general practice, flow charts are too cumbersome for large programs and so they are mainly used for isolated sub-routines, etc. We use pseudo code all the time. The actual pseudo code can become documentation for the written program by converting the pseudo code statements to programming comments.

Note that we are refining our pseudo code here by adding the **class** and **main** sections and declaring the variables that we will use in our program and specifying their type (**num, boolean, or String**).

This will transition us nicely into programming with java.

Class is the outer most container element for the pseudo code. You will see as we get into java that a class is implemented as a file. The name of the class will match the name of the java file: MyClass will be in MyClass.java.

There are two types of classes. The **main class** is the class that contains the special method (called **main**) that is run when the program is executed. Every java program must have a main class. A **model** or **object** class defines objects that we can use in our programs. You can't run a model class. (We will do more with model classes later in the next course. For now, when we start java we will only have main classes.)

Note the pseudo code syntax. The entire program is in a container **class … end class**. In a main class, the main method is **main() … return** and all the code statements for the program are located there. The main() … return block corresponds to and replaces the **start** and **end** statements that we used previously and the entire block is now in a class. Note how we indent our blocks.

```
class ClassName
  main()  // this was start
    //code goes here for our programs
  return  // this was end
```

end class

Also, let's begin to further refine our pseudo code to be specific about communication with our program users. So, every time we get input we will prompt the user like this. (I've actually had you do that from the beginning!) :

num birthMonth // declare a variable called birthMonth and specify that it holds a number
output "Enter your birth month [1 – 12] " // prompt the user for the input
input birthMonth // get the input and store it in birthMonth

The pseudo code on the slides occasionally to save space will do multiple inputs on one line like this:
input birthDay, birthMonth, birthYear
DON"T DO THIS!

Do one input per line and have a separate prompt for each. This is how you have to get the inputs in the Java code so it will help to do it this way in the pseudocode.

Here is an example of the code we did previously on the board in class for doubling a number using our new pseudo code conventions:

```
class DoubleMyFun
  main()
    // Declare variables  (this is a comment)
    num valueToDouble
    num doubledValue
    // Prompt and input
    output "Enter the number you want to double"
    input valueToDouble
     // Process
    doubledValue = valueToDouble * 2
     // Output
    output "The value you entered " + valueToDouble + " doubled is " + doubledValue
  return
end class
```

Variables:

We add some more details to our pseudocode and include variable declarations.

num yearOfBirth // a number variable
String ynChoice // a String variable to hold the user response to 'Yes or No?'
boolean isRaining // a boolean which is either true or false

We put our declarations at the top of the block immediately after the main() (see in the example pseudocode immediately above!)

Note that you only declare the type of the variable one time when you declare it. After that you only refer to the variable by its name:

num myAge = 61

print myAge

Mentioning the type again is an error because you are re-declaring a variable that already exists.

Conditional Structures:

So far, our programs have been pretty simple, now we are going to add conditional processing to our skill set.

Simple If:

```
if   BOOLEAN EXPRESSION then
   statement(s)
end if
```

If Then Else:

```
if   BOOLEAN EXPRESSION then
   statement(s)
else
   alternate statement(s)
end if
```

Cascaded If:

```
if   BOOLEAN EXPRESSION then
   statement(s)
else if BOOLEAN EXPRESSION then
   alternate statement(s)
else if BOOLEAN EXPRESSION then
   alternate statement(s)
else if BOOLEAN EXPRESSION then
   alternate statement(s)
…..
```

else // see how there is no test here? This is every other case.
alternate statement(s)

end if // or endIf

Lab:

1. Just submit this document (See directions below.) and insert your work here.

2. For each of the following tasks, provide the complete pseudo code. Include the new pseudo code elements: class … end class, main … return, user prompts, and declare all variables that the program needs. **You don't have to give me the Flow Charts except for one case indicated below!** Declare String variables with **String** (capital S) since that's what we do in java. We also have **num** and **boolean** variables in pseudocode. Declare all the variables at the top of the code block within main()… return. That is immediately after main().

Task 1 (2 pt): A program that converts a temperature in F that the user provides and returns the equivalent temperature in C. Hint: Google is your friend! Given F, solve for C. This program does not require an if structure and has a straight forward input – process – output structure!

```
class TemperatureConverter

  main()

        // Declarations

           num fahrenheit

           num celsius


        output "Enter the temperature in Fahrenheit:"


        input fahrenheit


        // Conversion formula

        celsius = (fahrenheit - 32) * 5 / 9


        output "Equivalent temperature in Celsius:", celsius


      return

    endClass
```

Task 2 (2 pts): As people pass through an entry kiosk at the theater, they are prompted to enter their age. If they are 21 or older, they get a paper wrist band. Code a logic program that asks the user to enter their age and then if they are 21 or over displays a message that they get a wrist band. (Note that the program does nothing if they are not 21 or over…)

```
class TheaterEntryKiosk

  main()
    // Declarations
    num age

    // Welcome message
    output "Welcome to the theater entry kiosk!"

    // Prompt user to enter age
    output "Please enter your age:"
    input age

    // Check age for eligibility
    if age >= 21 then
      // If 21 or older, output a message about getting a wristband
      output "Congratulations! You get a wristband."
    else
      // If younger than 21, output a different message
      output "Sorry, you must be 21 or older to get a wristband."
    end if

    // Thank you message
    output "Thank you for visiting the theater."

    // End main()
    return

endClass TheaterEntryKiosk
```

Task 3 (3 pts) : an application program where the user enters the price of an item and the program computes shipping costs. If the item price is $100 or more, then shipping is free otherwise it is 2% of the price. The program should output the shipping cost and the total price.

Submit the FlowChart for this Task 3 along with the pseudocode.

```
class ShippingCalculator


  main()
    // Declarations
    num itemPrice
    num shippingCost
    num totalPrice
    num FREE_SHIPPING_THRESHOLD = 100
    num SHIPPING_PERCENTAGE = 0.02

    // Prompt user to enter the price of the item
    output "Enter the price of the item:"
    input itemPrice

    // Check if the item price qualifies for free shipping
    if itemPrice >= FREE_SHIPPING_THRESHOLD then
      // If $100 or more, shipping is free
      shippingCost = 0
    else
      // If less than $100, calculate shipping cost (2% of the item price)
      shippingCost = itemPrice * SHIPPING_PERCENTAGE
```

```
    end if

    // Calculate total price
    totalPrice = itemPrice + shippingCost

    // Output shipping cost and total price
    output "Shipping cost: $", shippingCost
    output "Total price (including shipping): $", totalPrice

    // End main()
    return
end class ShippingCalculator
```

(Start)

|

v

[Input: Enter the price of the item]

|

v

[Decision: Is the item price $100 or more?]

**/ **

v (Yes) v (No)

[Process: Set shipping cost to $0] [Process: Calculate shipping cost (2% of item price)]

| |

v v

[Process: Calculate total price] *[Output: Shipping cost and total price]*

|

v

(End)

Task 4 (3pts): A program that asks the user to enter their birth month (integer 1 – 12 inclusive). If the user enters a value in range, the program echoes the input (“Your birth month is: N”) but if the value is not in the range it outputs an error msg (“You entered an incorrect month value: N”). Assume that the user can only enter numbers here.

```
class BirthMonthChecker


  main()
    // Declarations
    num birthMonth
    num MIN_MONTH = 1
    num MAX_MONTH = 12

    // Prompt user to enter their birth month
    output "Enter your birth month (1 - 12):"
    input birthMonth

    // Check if the entered value is within the valid range
    if birthMonth >= MIN_MONTH and birthMonth <= MAX_MONTH then
      // If in range, echo the input
      output "Your birth month is:", birthMonth
    else
      // If not in range, output an error message
      output "You entered an incorrect month value:", birthMonth
    end if
end class BirthMonthChecker
```

Task 5 (3 pts): (This task uses Strings and an if..then..elseif cascade) A program that prompts the user for their party affiliation (Democrat, Republican, or Independent) and responds accordingly with a Donkey, Elephant, or Man. (i.e. “You get a Democratic Donkey.”)

```
class PartySymbolGenerator
  main()
    // Declarations
    String partyAffiliation

    // Prompt user for their party affiliation
    output "Enter your party affiliation (Democrat, Republican, or Independent):"
    input partyAffiliation

    // Check party affiliation and respond accordingly
    if partyAffiliation equals "Democrat" then
      output "You get a Democratic Donkey."
    elseif partyAffiliation equals "Republican" then
      output "You get a Republican Elephant."
    elseif partyAffiliation equals "Independent" then
      output "You get an Independent Man."
    else
      // Handle invalid input
      output "Invalid party affiliation entered."
    end if
    return
end class PartySymbolGenerator
```

Submitting your work: carefully check your work. Rename your word file as **Lab02_Lastname_Firstname.docx** using your name. Submit this file using the Canvas assignment mechanism. Submit the exact same file a second time using the additional Canvas link for the extra credit option.