# *Programming I (420-B10-HR)*

# *Lab 15 – The do-while Loop*

Date assigned: Tuesday, December 8, 2015

Date due: **Tuesday, December 8, 2015**

**Learning Objectives**

Upon successful completion of this lab exercise, the student will be able to:

1. use a **do-while** loop in a program to:
   1. execute a body of statements a fixed number of times
   2. execute a body of statements using a sentinel
   3. accumulate sums and counts in a loop
   4. validate input data
2. Use flags to determine when to end a loop.

**Commands Used:**

**do**

*block of statements to be executed while the condition is* true

**while** **(***condition***);**

**To Be Handed In:**

1. The **Lab 15 Review Quiz** should be completed in Moodle.
2. The files ***username*\_B10\_L15\_do\_while** folder should be zipped and uploaded to **Moodle**.

**To Start:**

2. Download the **B10\_L15\_do\_while.zip** file and unzip it to your **H:\420-B10\Labs** folder. Rename it to ***username*\_B10\_L15\_do\_while**.

4. Start **Eclipse**.

5. Create a new **Java Project** called ***username*\_B10\_L15\_do\_while**.

# A Counting do loop

***Purpose:***Use a counting loop to input a fixed number of lines of data

***To Do****:*

## Open **Lab15A**. Run the program. Use any 5 values as input.

*Question:*

How many numbers are read in? \_5\_\_\_\_\_\_\_\_

## Change the **while** loop to a **do...while** loop:

do

{

number = input.nextInt();

System.out.println("Count is " + count + ". Number is " + number + ".");

sum += number;

product \*= number;

++count;

} while (count < 5);

*Question:*

How many numbers are read in? \_\_5\_\_\_\_\_\_\_

Which lines of the program are repeated (i.e. which lines are in the loop body)?

\_\_\_number = input.nextInt();\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

System.***out***.println("Count is " + count + ". Number is " + number + ".");

sum += number;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

product \*= number;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

++count;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Which line in the program marks the end of the body of the loop?

\_\_\_\_\_count ++;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***To Do:***

## Modify the program to work for **N** numbers, where **N** is input before the loop is executed. Test your changes. A sample dialog at the beginning of the program might look like this:

**How many numbers do you want to enter? 3**

**Enter 3 values, one per line**

*Question:*

What happens if you enter 0 for number of records to read?

\_\_\_because it’s a do while loop it still reads through the loop once\_\_\_\_\_\_\_\_\_\_\_\_\_

# A do loop using a Sentinel Value

***Purpose:*** Use a loop to input a number of values while a sentinel value is encountered.

***To Do:***

## Open **Lab15B**. Run the program. Use any 5 values as input. Enter -999 as the sixth number.

*Questions:*

What is the last number read? \_\_\_\_-999\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

What is the sum? \_25\_\_\_\_\_\_\_\_\_ What is the product? \_\_3125\_\_\_\_\_\_\_\_

Is the last number read included in the sum? \_\_no\_\_\_\_\_\_\_\_

## Run the program again. Enter -999 as the first number.

*Questions:*

What happens if -999 is the first number you enter?

\_\_\_All the values become 0 other than the product which is 1\_\_\_\_\_\_\_\_\_

What is the sum? \_0\_\_\_\_\_\_\_\_\_ What is the product? \_\_1\_\_\_\_\_\_\_\_

What is the minimum number of records that can be read with a **while** loop?

\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Modify the program to use a do-while loop instead of a while loop. Run the program and enter 5 values followed by -999.

*Questions:*

Does the do-while loop give the same results as the while loop when 5 numbers are input? \_\_Yes, the output is the same\_\_\_\_\_

## Run the program again and enter -999 as the only entry.

*Questions:*

Does the do-while loop give the same results as the while loop when -999 is the only number input? \_\_\_no\_\_\_\_\_\_\_\_\_\_\_

What happens when -999 is the first number you enter?

\_\_-999 defaults to zero\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

What is the sum? \_\_5\_\_\_\_\_\_\_\_ What is the product? \_\_\_5\_\_\_\_\_\_\_

What is the minimum number of records that can be read with a **do** **while** loop?

\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***To Do:***

## Modify the program to calculate and print the average with 2 decimal places. Test your changes.

# Using a Flag controlled loop to validate input data

***Purpose:*** Use a loop to validate a value entered from the keyboard.

***To Do:***

## Open **Lab15C**.The **readAge()** method in the class is incomplete. The code necessary to test for a valid age and terminate the loop is missing. A valid age is between 1 and 100. The error message should only be printed if the age is not valid. And the loop should only continue if the age is not valid. A **boolean** variable called **validAge** has been declared and initialized to false.

A **boolean** variable may be assigned to any valid **boolean** expression. For example:

boolean youngMale = (sex == 'm' && age <= 25);

will result in the following assignments

|  |  |  |
| --- | --- | --- |
| sex | **age** | **youngMale** |
| 'm' | 18 | true |
| 'm' | 30 | false |
| 'f' | 18 | false |
| 'f' | 30 | false |

Comments have been included in the method to indicate where the necessary code should be added.

### Assign **validAge** to an appropriate **boolean** expression. (A valid age is between 0 and 100).

### Add an **if** statement with a condition that tests **validAge** and executes the println() statement if **validAge** is false. (i.e. The age is not valid.)

### Modify the while clause to continue the while loop if **validAge** is false.

## Run the program to test your changes. Test for ages 102, -2 and your own age.

## Modify the program to ask the user to confirm the age if it is greater than 100. If the user confirms the age, accept it. If the user does not, prompt again. Use the char variable **confirm** to enter the answer to the confirm prompt.Two sample runs follow:

Enter an age between 0 and 100: 105

105 is awfully old. Is this the correct age? y

105 is a valid age. Thank-you

Enter an age between 0 and 100: 105

105 is awfully old. Is this the correct age? n

105 is not between 0 and 100.Please Re-enter: 95

95 is a valid age. Thank-you

# Loop Practice

***Purpose:*** Use a loop to display a range of prime numbers in a frame.

***To Do:***

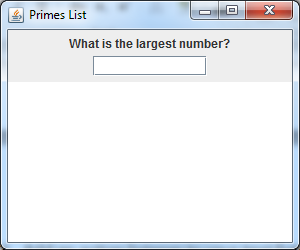
Atlas Computer Security has contacted you to write program that will provide a list of prime numbers less than a specified number.

## Create a class called **Primes** in the **primes** package. It should have a single method called **isPrime()** that returns true if the integer parameter is a prime number and false otherwise**.** The IPO for the **isPrime()** method is given here.

**isPrime() method IPO Diagram:**

| **Input** | **Process** | **Output** |
| --- | --- | --- |
| n – a number | Declare boolean notDivisibleYet  Declare numeric k   1. Initialize notDivisibleYet to true 2. Initialize k to 2 3. loop while (notDivisibleYet and k < n)    1. if n is divisible by k then       1. set notDivisibleYet to false   end if   * 1. increment k   end loop   1. return notDivisibleYet | true if n is prime; false otherwise |

## Create a frame called **PrimesListFrame**. Create a JPanel for the label and text field and put the panel in the center. Put the display area in the south. Set the size of the frame to (300,250) and your display area to (10,20). Your frame should look like:



## Add an action listener to your text field. Code the **actionPerformed()** method using the following algorithm. Use a **do while** loop instead of a **while** loop in your method.

**actionPerformed() method IPO Diagram:**

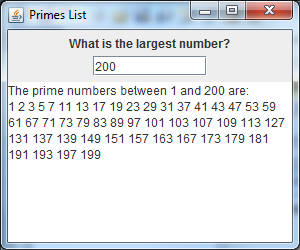
| **Input** | **Process** | **Output** |
| --- | --- | --- |
| none | Declare and instantiate a private **Primes** object called **primes**  Declare numeric maxNum, k   1. Get the number in the textfield and store it in maxNum 2. In the text area, display " The prime numbers between 1 and "   + maxNum + " are: "   1. Initialize k to 1 2. loop while k <= maxNum    1. if primes.isPrime(k) then       1. display k   end if   * 1. k = k + 1   end loop | none |

## Test your program using 200. Notice that the text doesn't wrap in the text area. To make your output display wrap around, call the following two methods in your constructor. (Substitute the name you have given to your JTextArea for *areaDisplay*.)

***areaDisplay*.setLineWrap(true);**

***areaDisplay*.setWrapStyleWord(true**);

## Test your program again. The output should look like:



# Review

***To Do:***

## Complete the **Lab 15 Review Quiz** on **Moodle**.