

Worksheet 3: Selection and Fault Finding

Updated: 9th March, 2020

The objectives of this practical is to:

- Practice selection statements (IF-ELSE, CASE)
- Building a Menu
- Finding faults in unfamiliar code
- Practice boolean expressions
- Tying it all together

Note: You must submit this practical worksheet **by the start of your next registered practical**, and have it signed off during that session.

Submissions must be done **electronically** through Blackboard. You are only required to submit once for this practical. You must submit everything you have completed in this practical, both in class and at home.

You must submit this Worksheet as a whole, so your weeks work **and** the Assignment Task, additionally if you wish you may include the optional task at the end.

You must also not modify the files in PDI\P03 until after your submission has been marked. During your practical your tutor will check the modified dates of the files using `ls -l (el)` before marking your work.

To get your **.bash_history** into a submittable format, first close all terminals down, using `<ctrl>-d`. Then open a new terminal, and type this command from anywhere:

```
[user@pc]$ history >~/Documents/PDI/P03/BashHistoryP03.txt
```

To create a gzipped tarball use the following command from your PDI folder:

```
[user@pc]$ tar -cvzf <studentID>_P03.tar.gz P03
```

For more information on what each argument of the above command does use:

```
[user@pc]$ man tar
```

Your submission will be marked during your practical session via an interview from your tutor comprised of a few questions.

Note that the questions asked in the interview may cover the entirety of the worksheet, not just the material in your Assignment Task. Your submitted work will be assessed and marks will be allocated accordingly.

1. Modifying Your Code

Last week you created a program that converted Celsius to Fahrenheit. Your first task is to allow the user to select the conversion that they want (i.e., Celsius to Fahrenheit or Fahrenheit to Celsius), this means we need to refactor our code.

First, make a copy of your **CelsToFaren.txt** file to your P03 directory. Also, copy the corresponding **.java** files over to your P03 directory.

Edit your **CelsToFaren** pseudocode to allow the user to select either Celsius ('C'/'c') or Fahrenheit ('F'/'f').

```
What temperature scale are you working with?  
  > (C)elsius  
  > (F)ahrenheit
```

Once the user has selected the appropriate conversion, they enter in their temperature, it is converted and output to the user. Now edit your **.java** to reflect these changes.

Do the same with your **CharConverter**, copy it to your P03. Now modify your code so that your program converts all lowercase to uppercase and vice versa.

Note: Ensure you only convert a-z to A-Z and A-Z to a-z, all other characters should print an error message to the user.

Now edit your **.java** to reflect these changes and make another copy of our testing document from our PDI folder into our P03 folder. Fill this out with your expected results first, then test your program to ensure it meets your requirements.

Warning: Make sure you change both the pseudo code and the java code as they must match at all times.

Once you have completed **both the pseudo and Java**, let your tutor know so they can have a look at it.

Note: You should only need to modify slightly, in order to achieve this.

2. What's on the Menu

Now that you've written all the necessary parts of your program, it's time to put them all together! In your P03 directory create a folder called **Menu** and change into it. Open a file called **Menu.txt** and write, in pseudocode, a menu.

This is an example of a menu that we expect to see in PDI.

```
Welcome to Programming Design and Implementation, Workshop 3
```

```
What would you like to do?
```

- > 1. Sum of 2 Integers
- > 2. Convert Temperature
- > 3. Convert a Character's case
- > 4. Print the ASCII equivalent of a Character
- > 5. Check if 2 Integers are Divisible
- > 6. Split a Date into its components
- > 0. Exit

Some things to consider when designing a menu:

- How will you get the user to enter their selection? Will you use a number, character, string or something else? Why are you considering this now? How will it affect your algorithm?
- Bare in mind usability, having a String for the selection means the user has to type the full string each time, and is prone to typo's. A Character or an Integer input is far more user-friendly.

When the program is executed the user should be prompted to make a selection (via input) from the available options.

If the user selects Exit then the program should print a good-bye message and exit with no further prompts. If another option is selected then the program should perform only that step. We may need to split apart some of the code that we have written.

Note: If you did not do the optional part of the practical last week, ignore option 6 for now, we will provide the functionality for it in the assignment task of this worksheet. Option 5 will be completed in part 4. Instead, for both, print out a message to the user that says: "This has not been implemented yet".

Once your pseudocode has been modified (and checked by your tutor) you can adjust your Java code to match. Remember to indent your code and update your documentation.

Note: If you are dealing with Integers for your menu input, you may assume that the user will only input Integers. If you are using Characters, you **must** handle any Characters that are input.

3. What IF I don't want a CASE?

Rewrite the pseudo code below changing the CASE statement to an IF statement.

```
OUTPUT "Please enter the direction you would like to go in
      (L) left, (R) right, (U) up, (D) down "
INPUT direction (Character)
CASE direction
    'L': 'l':
        OUTPUT "Going west"
    'R': 'r':
        OUTPUT "Going east"
    'U': 'u':
        OUTPUT "Going up"
    'D': 'd':
        OUTPUT "Going down"
    DEFAULT:
        OUTPUT "Going nowhere"
END CASE
```

Write this conversion in a file called **CaseToIf.txt**

4. Division Checker

Implement in pseudocode, using the practices you have learned in class, a program that inputs two (Integer) numbers, **numOne** and **numTwo**, and checks if the first number is divisible by the second number.

Note: You must write this within your **Menu** file, under Option 5.

If the two numbers are divisible then your program should output "Divisible" to the user, otherwise it should output "Not Divisible".

Note: Think about what logic errors you may have to account for in your testing.

Once you have completed the pseudocode make sure you get it checked. Once it has been checked you can implement your algorithm in Java.

5. Crafting Boolean Expressions

Given the following variable declarations (data types) craft Java boolean expressions for the following statements:

```
public static final TOL = 0.0001;  
String susan, jane;  
double steve, tony;  
int luke, scott, lucas;  
char john, matthew;
```

Write your answers in a file called **BooleanExpressions.txt**

Note: You will need to make use of `.equals()`, `.charAt()`, Real number equivalence and logic operators.

Convert the following statements:

- (a) Deciding if **susan** is the same as "sarah"
- (b) Deciding if **lucas** is an even number and **scott** is an odd number
- (c) Deciding if **john** or **matthew** are equal to the letter 'S'
- (d) Deciding if **jane** starts with letter 'V'
- (e) Deciding if **john** is the same as **matthew**
- (f) Deciding if **steve** is the same as **tony**
- (g) Deciding if **luke** is positive or **steve** is between 25.0 and 125.0 (inclusive)

Warning: Be mindful of mixed mode arithmetic!

Check your answers with your tutor. It is easy to craft a statement that look correct, but does not function correctly under all circumstances. You could also write a test program to see if your statements work.

6. Finding Faults in External Code

Your task is to download **FixThis.java** from Blackboard and modify it so it compiles and does as is expected, in its documentation by the author.

Note: There is more than 1 error within the program, however, the author does not know how to fix them and has asked for your help.

Ensure you test your program with the Testing file. (**FixThis_Test.txt**)

7. Assignment Task: Split My Date - Full Version

Your last task for this practical is to complete the following task **in your own time**.

Warning: This question goes towards your portfolio/assignment mark and thus any collusion will be dealt with as per university policy.

Last week, we gave you an optional task to implement a way of splitting a date into a series of numbers using **DIV** and **MOD**, those of you who completed the work copy your program into P03, we will be extending on that using selection. Everyone else, we will be implementing the full version this week.

Just to recap:

Your task is to create a program called **SplitMyDate** in pseudo code to convert an input from the user (**DDMMYYYY**) to their respective **day**, **month** and **year** variables. With the output formatted as below.

INPUT: 20022020

It is day 20 of month 2, in the year 2020.
9 years ago was day 20 of month 2, in the year 2011.

This week, we need to edit this so that it will display the suffix of the **day** (i.e., 1st), name of the **month**, and validate that the date exists. Future dates are acceptable.

Warning: You must use **DIV** and **MOD** for this and you may not use any external libraries to achieve this. You must write the algorithm yourself. You must only use the tools that we have learnt thus far (up to and including this week).

INPUT: 29022019

Error, this day does not exist.

INPUT: 29022020

It is the 29th day of Feburary, in 2020. It is a Leap year.

INPUT: 31122019

It is the 31st day of December, in 2019.

Note: You may need to use a combination of **CASE** and **IF-THEN-ELSE** to achieve this.

Hint: You may assume that the user will only ever enter an 8 digit Integer.

Once you have written it in pseudocode, convert it to Java and test it fully, this the includes **SplitMyDate_Test.txt** testing file.

End of Worksheet