Lab 4 – Flow Control

Aim

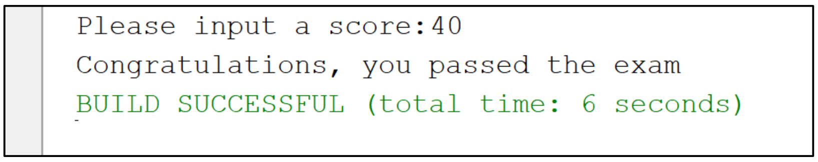
This lab aims to introduce you to flow control, including if-else statements and while loops. This material builds on material from weeks 1 and 2, so it is useful to have completed these weeks first. This practical will assume you have more knowledge of Java, and therefore, you will need to use your week 1 and 2 material

Tips

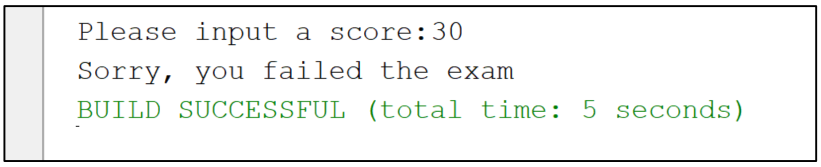
Remember, there are live BBB lab sessions with TAs who are there to answer your questions, so make sure you attend the session to ask and discuss the lab materials and exercises

If-Else Statements

- We will create a new class PassOrNot, which can judge if you have passed the exam
- If your input score is greater than or equal to 40, then print out “Congratulations, you passed the exam”
- If it is less than 40, print out “Sorry, you failed the exam”
- Your output might look like:



```
Please input a score:40
Congratulations, you passed the exam
BUILD SUCCESSFUL (total time: 6 seconds)
```



```
Please input a score:30
Sorry, you failed the exam
BUILD SUCCESSFUL (total time: 5 seconds)
```

- First, use two if statements to complete the question above
- Second, try to use the if-else statement to complete the same question

Created Autumn 2017, Modified Fall 2022
By Erick Purwanto, and Teng Ma

Nested If Statements – Valid Day

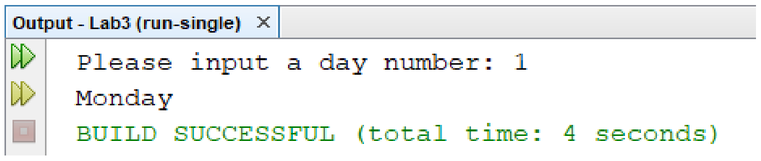
- We want to write a Java program that translate user integer input into its corresponding day (1 -> Monday, ..., 7 -> Sunday)

- However, the user may enter invalid number that has no corresponding name of days, so we first need to check whether the input number is a valid day number

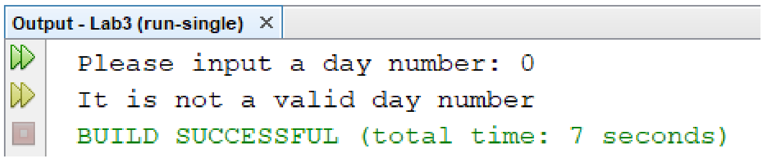
- To do this, we want to use nested ifs, i.e. ifs statement within another if statements. For the problem above, you may want to produce something following the model of:

```
if (validity condition is true) {
        if (variable is value1) {
                display output;
        }
        else if (variable is value2) {
                display output;
        }
        . . .
        else {
                display output;
        }
}
else
        display output;
```

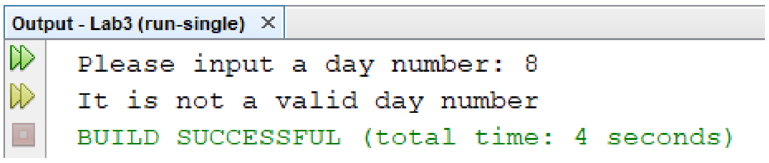
- Some examples of the console output is shown below


Output - Lab3 (run-single)

```
Please input a day number: 1
Monday
BUILD SUCCESSFUL (total time: 4 seconds)
```


Output - Lab3 (run-single)

```
Please input a day number: 0
It is not a valid day number
BUILD SUCCESSFUL (total time: 7 seconds)
```


Output - Lab3 (run-single)

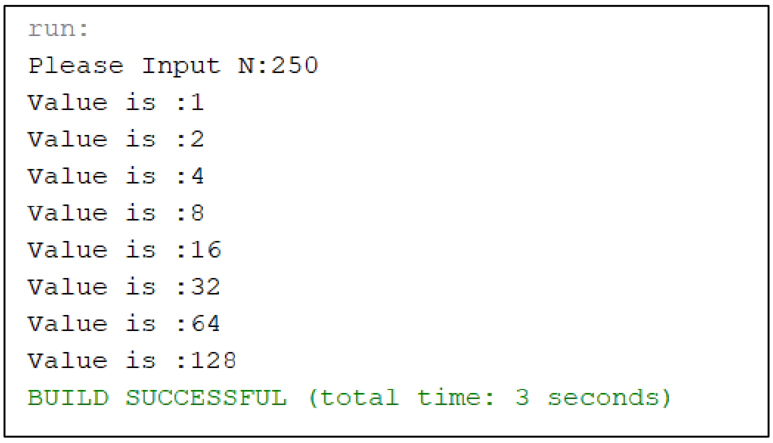
```
Please input a day number: 8
It is not a valid day number
BUILD SUCCESSFUL (total time: 4 seconds)
```

Indefinite Loops – Powers of Two Version Two

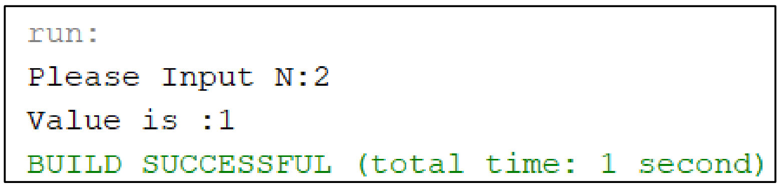
If we want to repeat something a number of times, a loop is very useful. You have learned will introduce the while loop in lecture, in particular using while loop to print n first powers of two.

While loop is particulary useful when we do not now how many times exactly we must loop. To illustrate this, we now write a Java program that takes an integer input n and prints all powers of two (starting from $2^0 = 1$) that is less than n

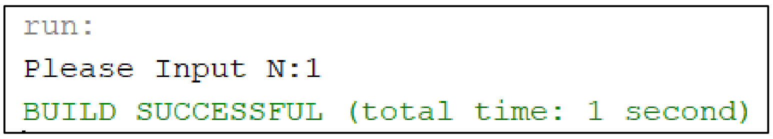
- Create a new class called PowersOfTwoV2 and its main method
- Write the code that accepts an integer input n from user as usual
- Initialize a variable val that will be used to compute the powers of two to 1
- Use a while loop to check whether val is less than n
    - If that is the case, we go to the code block and print val to the console
    - We also need to update val and multiply it with two
    - This while loop will eventually end since the value of val will be doubled to finally exceed n
- Some examples of the console output is shown below



```
run:
Please Input N:250
Value is :1
Value is :2
Value is :4
Value is :8
Value is :16
Value is :32
Value is :64
Value is :128
BUILD SUCCESSFUL (total time: 3 seconds)
```



```
run:
Please Input N:2
Value is :1
BUILD SUCCESSFUL (total time: 1 second)
```



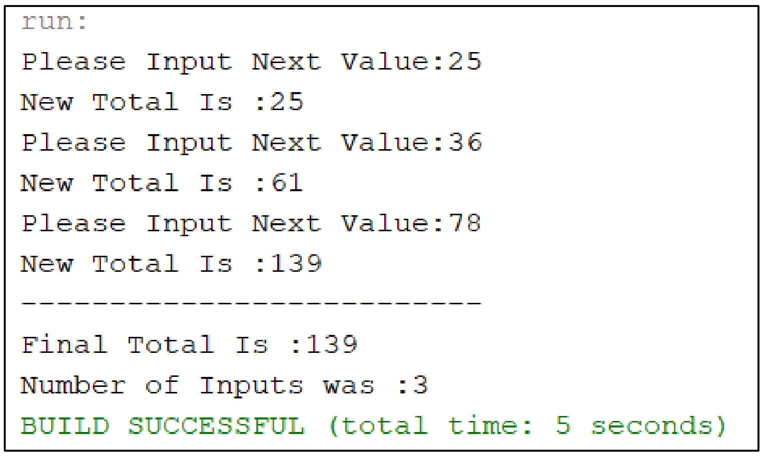
```
run:
Please Input N:1
BUILD SUCCESSFUL (total time: 1 second)
```

While Loops – Variable handling

As seen above, while loop is very useful for a number of things, not just counting. Let us have one more practice with this style of computation.

In this problem, we want to keeping asking and adding user input numbers until it exceeds a certain threshold, in this case, 100

- Create a new class SumOneHundredOrMore
- Your program will take an integer input from the keyboard. We want to add each integer to a total and display both an ongoing total and then the final total
- To do this, we need an initial integer total variable set to 0, and then a loop where user input is repeatedly asked. This input is added to the total. The loop only stops when the total is greater than 100
- Using the code you have written in this lab so far, and your lecture notes, try to design and write a program that will solve this problem
- After the loop finishes, you should display both the total and the number of times the program looped
- An example of the console output is shown below



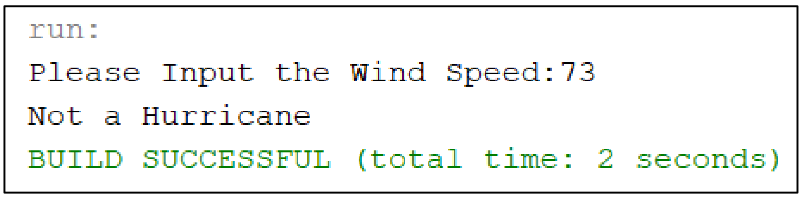
```
run:
Please Input Next Value:25
New Total Is :25
Please Input Next Value:36
New Total Is :61
Please Input Next Value:78
New Total Is :139
--------------------------
Final Total Is :139
Number of Inputs was :3
BUILD SUCCESSFUL (total time: 5 seconds)
```

Lab Exercise 4.1 Nested Ifs

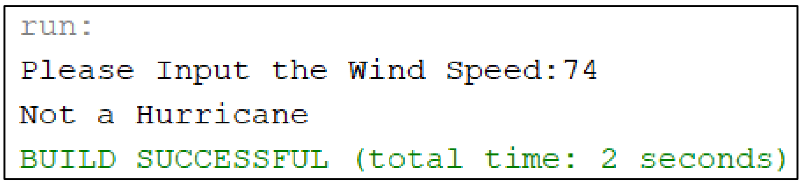
- Write a Java program Hurricane.java that that takes an input integer wind speed and reports whether it is as a hurricane, and if so, whether it is a Category 1, 2, 3, 4, or 5 hurricane, according to the Saffir-Simpson Scale.
- The table of the wind speed and hurricane category according to the Saffir-Simpson Scale is found below:

Category	Wind Speed (miles per hour)
1	74 – 95
2	96 – 110
3	111 – 130
4	131 – 155
5	156 and above

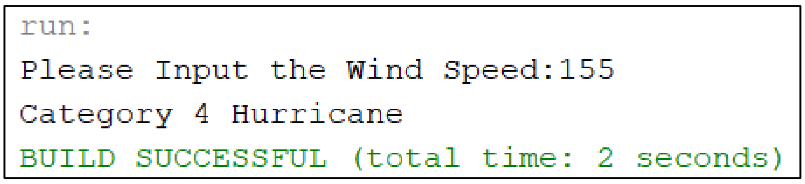
- Your program’s input/output interaction should look like:



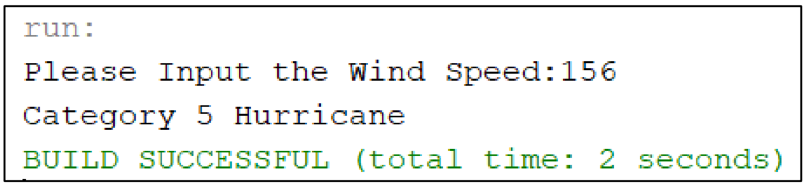
```
run:
Please Input the Wind Speed:73
Not a Hurricane
BUILD SUCCESSFUL (total time: 2 seconds)
```



```
run:
Please Input the Wind Speed:74
Not a Hurricane
BUILD SUCCESSFUL (total time: 2 seconds)
```



```
run:
Please Input the Wind Speed:155
Category 4 Hurricane
BUILD SUCCESSFUL (total time: 2 seconds)
```



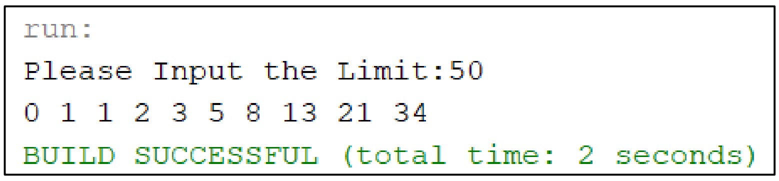
```
run:
Please Input the Wind Speed:156
Category 5 Hurricane
BUILD SUCCESSFUL (total time: 2 seconds)
```

Lab Exercise 4.2 Indefinite Loop with While

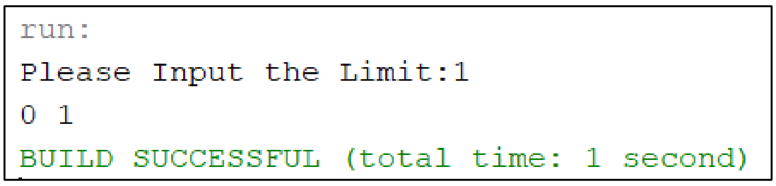
- The Fibonacci Sequence is the series of numbers: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...
- It starts with 0 and 1
- The next number is found by adding up the two numbers before it
    - The third number 1 is found by adding the two numbers before it, 0+1
    - The fourth number 2 is found by adding the two numbers before it, 1+1
    - The fifth number 3 is 1+2, and so on

- Write a Java program that takes an input *positive integer*, n, from the user, and prints all numbers in the Fibonacci sequence that is less than n

- Your program's input/output interaction should look like:



```
run:
Please Input the Limit:50
0 1 1 2 3 5 8 13 21 34
BUILD SUCCESSFUL (total time: 2 seconds)
```



```
run:
Please Input the Limit:1
0 1
BUILD SUCCESSFUL (total time: 1 second)
```

This is the end of CPT111 2022 Lab 4 Task Sheet.

Created Autumn 2017, Modified Fall 2022
By Erick Purwanto, and Teng Ma