Recitation 3

September 23, 2021

Topics to Recap

- Conditionals
    - Syntax (&&, ||, and ! instead of &, |, and "not")
    - Truth tables
    - Optional: if/else if/else chains and switch() statements
- Functions with return values
    - When to use **void** return type
    - Nesting functions with non-void return types, i.e. triple(squared(5.1))
- The `this` keyword

Examples for Recap

1. Create a class `Professor` with a function `willTeachToday(int age, String weather, boolean feelsLikeIt)` that returns `false` if any of the following are true:
    - `feelsLikeIt` is `false`,
    - `age` is greater than `70`,
    - `weather` is equal to `"rainy"` or `"snowing"` and `age` is greater than `55`

And returns `true` otherwise

3. Determine if each of the following functions would compile. Assume they are defined within the following class:

```
public class Rec3Example {
  private int pi = 3;

  /* functions would be here... */
}
```

```
public void makeSomePi(int pi) {
  this.pi = pi;
}

public double improvePi(boolean better) {
  if (better OR !better) {
    return 3.14;
  }
}

public void moarPi(int smallerDigits) {
  System.out.print(this.pi + "." + smallerDigits);
}

public int stopUsingPiPlease(boolean prettyPlease) {
  if (prettyPlease) {
    System.out.println("Ok fine");
  } else {
    System.out.println("No");
  }
}
```

Recitation Problem Set

GROUPS:

Group				
Group 1	Zhang, Minzheng	Thenappan, Bala Sundar	Tims, George	Cruz, Marye I
Group 2	Mammadov, Elmar	Xue, Mingxin	Lai, Qimei	He, Ziyi
Group 3	Chou, Randy	Ye, Huifang	Carnation, Kayla Rae	Yu, Qingyu
Group 4	Chheda, Shagun Pritesh	Zhang, Miaoyan	Huang, Wenyi	Xiao, Zijian
Group 5	Zhang, Yang	Gallagher, John Manus	Richmond, Christian	Bales, Elijah
Group 6	Kong, Rachel	Patel, Rishi	Ng, Wai Chung	Jiang, Yao
Group 7	Nojoomi, Radin	Biscaro, Denise	Wu, Jeng-Ru	Bernat, Kevin Bruno
Group 8	Arguello-Gonzalez, Marcos Abraham	Schnall, Aaron Hewitt	Wang, Kehan	Li, Yunhe
Group 9	Liu, Jiayun	Cho, Suebin Grace	Guo, Zhaosen	Liu, Shufan
Group 10	Chen, Xiyue	Lee, Jaeyoung	Liu, Xinyue	Cai, Jialin
Group 11	Hu, Lucy Qian	Wang, Yuanqi	Kallas Jatene, Rafael	Sheng, Xinyue
Group 12	Ren, Yue	Zhang, Yihong	Cheema, Sardar Asfandy	Chen, Zheyi
Group 13	Qiu, Xi	Shah, Rushabh	Choi, Jae Ho	Kim, Yunchae
Group 14	Graham, Alexander Richard	Nguyen, Tai D	Zhang, Zhihui	Wang, An-Jie
Group 15	Pizzico, Tyler R	Qiu, Chengzhuo	Rigas, Andrew	Hu, Yuxin
Group 16	Sabri, Rita	Pinheiro, Benjamin B	Williams, Levester Randall	Zhang, Han
Group 17	Pace, Benjamin Michael	He, Donglun	Lim, Xi Zhen	Patel, Siddharth Bhagwanji
Group 18	Sha, Yumeng	Kung, Ling-Hsin	Yiu, Hon-Cheung	Wang, Liujia

1. Create a class that mimics the sorting hat from Harry Potter. It will begin by asking the user for a `name`, then asking for a `hair color`, and finally printing out "`name` will be placed in the `houseName` house", where `houseName` is determined by the following rules:

- Any user whose `name` begins with the letter A, B, or C (case insensitive) will be placed in the Hufflepuff house. - Any user whose `name` begins with the letter H or R (case insensitive) will be placed in the Griffindor house, unless that user also has a `hair color` of "blonde", in which case the user is sent to the Slytherin House. - Any user whose `name` does *not* begin with any of the above letters (A, B, C, H, or R, case insensitive) will be placed in the Ravenclaw house if they have a `hair color` of "brown" or "black", and will be placed in Slytherin otherwise. - The only possible houses (values of `houseName`) are Griffindor, Hufflepuff, Ravenclaw, and Slytherin.

2. Complete lab problems 5.18.2 and 5.18.3 from the zybook. You can work in the textbook to get instant feedback, although **you'll need to submit your solution to Gradescope either way**. I've copied the problems below:

- 5.18.2:

> Create a conditional expression that evaluates to string "negative" if userVal is less than 0, and "non-negative" otherwise. Ex: If userVal is -9, output is: `-9 is negative.`

```
import java.util.Scanner;

public class NegativeOrPositive {
   public static void main (String [] args) {
      Scanner scnr = new Scanner(System.in);
      String condStr;
      int userVal;

      userVal = scnr.nextInt();

      condStr = /* Your solution goes here  */;

      System.out.println(userVal + " is " + condStr + ".");
   }
}
```

- 5.18.3:

> Using a conditional expression, write a statement that increments numUsers if updateDirection is 1, otherwise decrements numUsers. Ex: if numUsers is 8 and updateDirection is 1, numUsers becomes 9; if updateDirection is 0, numUsers becomes 7.
>
> Hint: Start with "numUsers = ...".

```
import java.util.Scanner;

public class UpdateNumberOfUsers {
   public static void main (String [] args) {
      Scanner scnr = new Scanner(System.in);
      int numUsers;
      int updateDirection;

      numUsers = scnr.nextInt();
      updateDirection = scnr.nextInt();

      /* Your solution goes here  */

      System.out.println("New value is: " + numUsers);
   }
}
```

3. **OPTIONAL CHALLENGE:** In Java, there is something called the **ternary operator**. This is an expression that returns one of two values based on the result of a conditional. The syntax is as follows: `result = (conditional) ? optionIfConditionalTrue : optionIfConditionalFalse;` . For example, after running the following code:

```
int x = 0;
String favoriteClass = "CIT591";
x = ("CIT592".equals(favoriteClass)) ? 500 : -27;
```

The result of `x` will be `-27` since `"CIT592".equals(favoriteClass)` evaluates to `false`. Using that knowledge, determine the output of the following function (*without* simply copy/pasting it into Eclipse or IntelliJ):

```
public int logicalConfusion() {
  int a = (True || False) ? 3 : 5;
  int b = (a > 4 && True) ? (a - 2) : (a * 2);
  boolean c = a - 5 > 0;
  b++;
  int theBigOne = ((c || (b - 4 == a)) && ((a * 2 - b >= 0) || !(!(c)))) ? (b - 7) : (a * -3); // Read this line carefully!
  return theBigOne;
}
```