recitation1.md

Recitation 1

September 9, 2021

Topics to Recap

- Using Eclipse
  - Creating a new project
  - Creating classes within a project
  - Running programs
- Hello World
  - Defining a class name
  - Writing the main method
- Building Blocks for Basic Programs
  - `System.out.println` vs. `System.out.print`
  - Declaring a variable
  - Assigning a value to a variable
  - Basic arithmetic (`+, -, *`)
- Gathering Program Input
  - Instantiating a new Scanner
  - Reading an int from the command line (`nextInt()`)
  - Reading the next line using `nextLine()`
    - Using `skip("\n")` to accept a new line after asking for an int.
- Objects and Classes
  - Defining an object as (internal data + valid operations)
  - Objects are high level views of complex constructions focus on what we can do with them
  - Classes define objects
    - `Thing.java` is the Class that specifies how `Things` are built and what you can do with `Things`

Examples for Recap

1. Write a program that asks for the user's age and THEN their name. Then, print out: `"A year from now, you'll be <AGE + 1>, <NAME>"` .

2. Describe how we can represent an MCIT student as a Java object.

   - What are the relevant internal data components?
   - What are the valid "operations?"

3. You're writing a program to represent a professional kitchen in a restaurant simulation video game. What objects might you need to manipulate in `Restaurant.java` , and what might you do with each of those objects?

   - For example, you might have a `Customer` object. These would represent the people who show up to eat. You can `seat` a `Customer` , you can `takeOrder` from them, a `Customer` can be `served` , and the `Customer` can `pay` .

Recitation Problem Set

> Submit your answers to the following questions to Gradescope. You'll be awarded points primarily on completeness, although some degree of correctness and effort is necessary to earn credit.

GROUPS:

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

> If members of your group are not present, that's OK. If you're alone, join another group. If you're working remotely, feel free to congregate and work together on gather.town.

1. Identify whether each program will compile. If it does not, explain why not.

   i.
   ```
   public class Example (
       public static void main(String[] args) (
           System.out.println("Example One");
       )
   )
   ```

   ii.
   ```
   public class Example {
       public static void main(String[] args) {
           System.out.println('Example One');
       }
   }
   ```

   iii.
   ```
   public class Example {
       public static void main(String[] args) {
           System.out.printIn("Example One");
   ```

```
    }
}
```

iv.

```
public class Example {
    public static void main(String[] args) {
        System.out.println(age + 4);
    }
}
```

v.

```
public class Example {
    public static void main(String[] args) {
        int age;
        System.out.println(age + 4);
    }
}
```

vi.

```
public class Example {
    public static void main(String[] args) {
        int age = 3.4;
        System.out.println(age + 4);
    }
}
```

vii.

```
public class Example {
    public static void main(String[] args) {
        int age = 3;
        System.out.println(3 + 4);
    }
}
```

2. Write a brief program that prompts the user for a number of days (as an `int`), and prints out that same time in numbers of hours, minutes, and seconds, followed by a new line.

   For example:

   ```
   input:
   4
   output:
   96 5760 345600
   ```

   Make sure to test your code on different numbers of days, and check your work manually. Does anything interesting happen when you provide a large number of days?

3. Write a brief program that prompts the user for a number of seconds (as an `int`). Print out how many full days go into that number of seconds. You'll need to know that division between two `ints` in Java (using the `/` operator) returns an integer result. This means that `5 / 2 == 2`. Since the true result is `2.5`, the decimal portion of this number is truncated and just `2` is the result.

   For example:

   ```
   input:
   345599
   output:
   3
   ```

4. **OPTIONAL CHALLENGE:** Write a program that prompts the user for a number of seconds (as an `int`). Print out how long this is in days + hours + minutes + seconds. To do this, you'll need to use the modulo operator (`%`), which returns the remainder after division. For example, `5 % 2 == 1`, since `2` goes into `5` twice, and `5 - (2 * 2) == 1`. We'll cover `%` more later, but you can attempt to use it to solve this problem.

   ```
   input:
   345599
   output:
   3 DAYS, 23 HOURS, 59 MINUTES, 59 SECONDS
   ```