Lab 5a

CSC-121, Fall 2025

Group Exercises

Q1. Reading and Writing Code

1. Find the ~~Erers~~ Errors

The following method snippets each have an error. Find and correct them.

1.

Java

```java
public static void sayHello();

{

    System.out.println("Hello");

}
```

2.

Java

```java
public static double timesTwo(double num)

{

    double result = num * 2;

}
```

3.

Java

```java
public static int half(double num)

{

    double result = num / 2.0;

    return result;

}
```

Q2. Deja Vu

The formula for converting a temperature from Fahrenheit to Celsius is:

$C = \frac{5}{9}(F - 32)$

…where F is the Fahrenheit temperature and C is the Celsius temperature. Write a method named celsius that accepts a Fahrenheit temperature as an argument. The method should return the temperature, converted to Celsius. Demonstrate the method by calling it in a loop that displays a table of the Fahrenheit temperatures 0 through 20 and their Celsius equivalents.

Partner Exercises

Work together with a partner to build the following programs.

Q1. Dictionaries / Maps

In Python, a dictionary is a datatype that maps unique keys to specific values. For example:

Python

```python
>>> myDictionary = {}
>>> myDictionary['apple'] = 1
>>> myDictionary['banana'] = 2
>>> myDictionary['cherry'] = 3
>>> myDictionary
{'apple':1, 'banana':2, 'cherry':3}
>>> myDictionary.keys()
dict_keys(['a', 'b', 'c'])
>>> myDictionary.values()
dict_values([1, 2, 3])
>>> myDictionary.get('banana')
2
>>> myDictionary.get('cherry')
3
```

We can associate symbolic keys (Like the fruit names above) with an entry in the dictionary and store a value there. Each key must be unique, so you could not have a second 'apple' key associated with a different value from the first 'apple'. The key and the value can be different types, like how above we have a String used for the key and a number for the value.

Dictionaries are built into Python as a primitive data type associated with the { } symbols. We have dictionaries in Java as well, but they are called "Maps" and are part of the standard library. To use them, we have to include the library and learn their unique Java syntax.

:

Java

```
import java.util.HashMap;
public class Sandbox {
    public static void main(String[] args)
    {
        HashMap<String,Integer> myMap = new HashMap<String,Integer>();
    }
}
```

First, we have to import the HashMap library into our project. Then, we can create a HashMap object. It looks a bit different than other objects we've made before, so let's look at what is new.

HashMap<String,Integer> myMap = new HashMap<String,Integer>();
The type of our variable is a hash map, but it's a specific type of hash map. Remember that Java is a strongly-typed language. Our maps can have different types for the keys and values, so we have to specify what those types are within the < > angle brackets. Special classes called "Templates" allow us to customize their types in this way. For this example, the keys will be Strings, and the values will be ints.

NOTE: We use the "Integer" type instead of just int because the data types supplied to the HashMap template must be classes. "Integer" is a "class wrapper" for the int primitive data type, and is very useful for cases like this. We've used it once before for the Integer.parseInt() method to convert a String into an int.

Now that we have created the map, let's learn how to use it to hold information for us. We're going to use the HashMap's "put" and "get" methods to store and retrieve data. Let's try replicating our earlier python example with fruit and numbers.

Java

```java
import java.util.HashMap;
public class Sandbox {
    public static void main(String[] args)
    {
        HashMap<String,Integer> myMap = new HashMap<String,Integer>();
        myMap.put("apple", 1);
        System.out.println("Value of apple: " + myMap.get("apple"));
    }
}
```

In the code above, we "put" the value of 1 under the key "apple". Now when we "get" the value of "apple" from the HashMap, we get 1. Let's try some more.

Java

```java
import java.util.HashMap;
public class Sandbox {
    public static void main(String[] args)
    {
        HashMap<String,Integer> myMap = new HashMap<String,Integer>();
        myMap.put("apple", 1);
        myMap.put("banana", 2);
        myMap.put("cherry", 3);
        System.out.println("Value of apple: " + myMap.get("apple"));
        System.out.println("Value of banana: " + myMap.get("banana"));
        System.out.println("Value of cherry: " + myMap.get("cherry"));
    }
}
```

In each case, we are able to print the value associated with each key that we have put into the table. Let's try another test.

Java

```java
import java.util.HashMap;
import java.util.Scanner;
public class Sandbox {
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        HashMap<String,Integer> myMap = new HashMap<String,Integer>();
        ??? // 1. Your "put" code here!

        while (true)
        {
            System.out.print("Enter a student's name: ");
            String name = scanner.nextLine();
            System.out.println();

            if (???) // 2. Your boolean condition here.
            {
                System.out.println(name + "'s age is " + ???); // 3. "get" here.
            } else {
                break;
            }
        }
        scanner.close();
    }
}
```

The program above is meant to ask for the name of a student and then print that student's age. When an unknown name is entered, the program ends. There are three spots marked with ??? that you need to replace with code to complete the program.

1. Add your name and your lab partner's names into myMap along with your respective ages. Your names are the keys, and your ages are the values.

2. You need to check if the entered name is in the HashMap. Look at the Java Documentation for HashMap and see if you can find a method that tells you if the HashMap *contains* a specific key.

3. Get and print the value from the HashMap for the entered student name.

Great job! Now that you have some practice with HashMaps, try taking another look at the pizza problem from last week's Lab 4c activity section. For reference, the full program is on the next page.

Java

```java
import java.io.File;
import java.io.FileNotFoundException;
import java.util.HashMap;
import java.util.Scanner;
public class Sandbox {
    public static void main(String[] args)
    {
        File csvFile = new File("lab_04c_pizza_time.csv");
        Scanner csvScanner = new Scanner(csvFile);
        double totalSales = 0.0;
        csvScanner.useDelimiter(",|\\r?\\n"); // Catch commas and newlines.
        csvScanner.nextLine(); // Skip the first row of column names.
        while (csvScanner.hasNext())
        {
            String name = csvScanner.next();
            String date = csvScanner.next();
            double price = Double.parseDouble(csvScanner.next());
            String pizzaType = csvScanner.next();
            totalSales += price;
            System.out.printf("Name: %s, price: $%.2f\n", name, price);
        }
        csvScanner.close();
        System.out.printf("Total sales: $%.2f\n", totalSales);
    }
}
```

The first question asked you to read all the spreadsheet data and sum up all of the pizza sales for the entire day. The second question asked you to sum up the sales for each individual employee and report each when asked. Now we can tackle that second question! Here's the tasks to do:

1. First, create a HashMap with employee name Strings as the keys and a Double for the values.

2. Let's sum up each employee's sales in the same loop where we currently sum up the day's total sales. Each time you read in a new row of data, save the name and the price. If the HashMap does not contain a key matching the employee's name, create an entry in the HashMap for it and set the sale price as the value. If an entry in the HashMap does exist for the employee, get the value from the HashMap under that employee name key, add that value to the price just read in from the csv, and then put the total in the HashMap under the employee's name key.

3. Similar to our student name/age program above, let's add a loop to the end of our pizza program where we ask the user to type the full name of an employee, and then we print the total sales of that employee.

Individual Tasks

1. Project Check-In

Have you started your Project **#2**, yet? ***Don't wait until the last minute!*** You should present your plan to one of the Tech Fellows and get feedback. Remember, part of your grade is based on doing the check-in.

2. CodePath Mid-Semester Survey

How do you feel that the semester has been going so far? Please let us know in the mid-semester survey. This will help the teaching staff by providing feedback on what we could do to make the course better.

Survey Link: https://www.surveymonkey.com/r/cir-fall25-mc