Short Answer Questions

Answer the following questions with complete sentences in your own words. You are encouraged to conduct your own research online or through other methods before answering the questions. If you research online, please consult multiple sources before you write down your answers. You are expected to be able to explain your answers in detail (Provide examples to each question).

1. What is the Stream API in Java 8? What are the differences between intermediate and terminal operations for Stream?
2. What is the Optional class in Java and why do we need it?
3. What is an exception and what is an error in Java? What are the differences between them?
4. What are the types of exceptions in Java? What are the differences?
5. How can we handle an exception in Java?
6. What is a custom exception and why do we need to use it?
7. What is the difference between the *final* and *finally* keywords?
8. Is the following code legal?
   Why?
   ```
   try {
   } finally {
   }
   ```
9. Is there anything wrong with the following exception handler as written? Will this code compile?
   ```
   try {

   } catch (Exception e) {

   } catch (ArithmeticException a) {

   }
   ```

13. Match each situation in the first list with an item in the second list.
    a. Scenarios
       i. int number = “four”;
       ii. public void recursion() { recursion(); }
       iii. A program is reading a stream and reaches the end of the stream marker.
       iv. You write code to read from a file but misspelled the filename
    b. Exceptions/Errors
       i. Error
       ii. checked exception
       iii. compile error
       iv. no exception

Coding Questions

Write code in Java to solve the following problems. Please write your own answers. You are highly encouraged to present more than one way to answer the questions. Please follow best practice when you write the code so that it would be easily readable, maintainable, and efficient. Clearly state your assumptions if you have any. You may discuss with your peers, but please write your own code.

1. Java 8 Stream API practices
    a. Create an Employee class with two instance fields: a String name and an Integer salary
    b. Create a List of employees (be sure to implement the equals and hashcode methods)
    c. Use Java 8 Stream and Lambda Expression to achieve the following:
        i. Calculate the average salary of the list of employees
        ii. Filter for employees whose salary exceeds 80000 and print out their names
        iii. Collect a Map of employees where the key is the name of employee (String) and the value is the employee object (Employee)
        iv. Find any Employee whose name starts with the character X
            1. If such an employee exists then print out the name, if not then print “no such employee exists”
        v. Create a String that contains the name of every single employee in the list concatenated together.

2. Create a user defined exception class called NonIntResultException which is generated when the result of dividing two integer values produces a result with a fractional component [i.e the result is not an integer]
    a. NonIntResultException contains:
        i. A constructor with parameter that represent the two integer values
        ii. Generates an appropriate message, for example if the two integers are 7 and 2, the resulting exception message would be: 7 divided by 2 is not an integer

3. Create the IntegerArrayMath class with integer division method:
    a. Loops thru instance field array and attempts to divide each value of number array by the corresponding value of denom instance field array. such as number[0]/denom[0] and number[1]/denom[1],etc
    b. If the result of the division is an integer, then print out a message indicating the result of the division such as 8/4 is 2.
    c. If the result of the division is not an integer, then throw and handle a NonIntResultException and continue processing the result of the number array elements.
    d. The method should, using exception handling also handle any attempt to divide by zero (arithmetic exception) the program should display an appropriate message and then continue processing the rest of the number array elements
    e. Assume both arrays are the same length.

4. Write a method to convert the given byte array back into a file:
[104, 116, 116, 112, 115, 58, 47, 47, 119, 119, 119, 46, 121, 111, 117, 116, 117, 98, 101, 46, 99, 111, 109, 47, 119, 97, 116, 99, 104, 63, 118, 61, 100, 81, 119, 52, 119, 57, 87, 103, 88, 99, 81]