Core Java

Task Sheet

General Instructions

You will have the afternoon to complete these tasks.

Create a new project for all of the following tasks. As a suggestion, you can call your main project “tasks” but you can call it whatever suits you.

Each task will go into its own separate package and will be run individually. Each task will require its own `main` method. This would mean the following structure with each `App.java` file having its own `main`.

```
|- com.sparta
        |
        |- day1
        |     |- debug
        |     |     |- DebugApp.java
        |     |
        |     |- calculator
        |           |- CalculatorApp.java
        |
        |- day2
```

For all tasks you should be able to explain and justify the solution that you arrive at.

Core Java

Tasks

A Simple Calculator

Package: com.sparta.day5.calculator

Class: CalculatorApp

Write a program to prompt the user for two positive integer values and then an operator which should be one of the following +, -, /, *. Your program should then output the full sum as text with the answer.

For example if the user enters 5 2 * your program should output:

```
Your sum is 5 * 2 and the answer is 10
```

Grid

Package: com.sparta.day5.grid

Class: GridApp

Write a program to prompt the user to enter 2 numbers: one for number of columns and one for number of rows and outputs a grid of asterisks with the specified number of rows and columns

Hint: You will need to use one loop nested inside another.

For example, given width = 3 and height = 4, the program should output the following grid:

```
* * *
* * *
* * *
* * *
```

Average of Positive and Negative Values

Package: com.sparta.day5.averageposneg

Class: AverageApp

Core Java

Write a program that prompts the user to enter 10 integer values, which can be positive or negative, and outputs 4 numbers:

- the sum of all the positive integers entered
- the average of all the positive integers entered
- the sum of all the negative integers entered
- the average of all the negative integers entered

If no positive numbers or no negative numbers were entered then just output an appropriate message instead.

Diamond

Package: com.sparta.day5.diamond

Class: DiamondApp

Write an application that uses loops to output a diamond where the width is defined by the user but should be no less than 2 and no more than 40. Your program should output an appropriate message if the width entered is outside of the permitted range.

For example, given a width of 4, the program should output the following:

```
   *
  * *
 * * *
* * * *
 * * *
  * *
   *
```

Determined Loops

Make a note in a text file whether each of the following is deterministic or non-deterministic (ie could be solved deterministically or non-deterministically) and justify your choice:

a) writing Christmas cards to family
b) washing up
c) reading a book
d) waiting for someone to answer the phone
e) counting to 10 in French
f) playing a game of 501 in darts
g) laying the table for a dinner party
h) answering this question

Core Java

Paper Rock Scissors

Package: com.sparta.day5.gameprs

Class: GameApp

Write a game of Paper, Rock, Scissors; where the user is prompted to enter their choice as a case-insensitive string (e.g. Paper or paper would be valid inputs) and the computer's choice is generated randomly.

The first to score 3 wins.

At the conclusion of the game your program should output the result (who won) and the scores of both players.

If you are not familiar with the game, you can learn a little about it here (https://en.wikipedia.org/wiki/Rock-paper-scissors).

Vowel Escape

Package: com.sparta.day5.vowelescape

Class: VowelApp

Write a program that allows the user to enter a string and returns two strings:

- all of the vowels in upper case
- all of the vowels removed

Find the Highest

Package: com.sparta.day5.highest

Class: HighestNumberApp

For the following list of integers:

```
List<Integer> numbers = new ArrayList<>(Arrays.asList(
    5, 12, 7, 19, 3, 42, 8, 16, 27, 33,
   4, 9, 21, 11, 6, 25, 14, 2, 18, 30
));
```

Write a method that will accept this as an argument and return the second highest number.

Core Java

Character Generator

Package: com.sparta.day5.character

Class: CharacterGeneratorApp

Write a program that generates five attributes for a character (randomly):

1. Strength
2. Intelligence
3. Wisdom
4. Dexterity
5. Constitution

All attributes should be in the range 3 to 18.

The character can be one of the following classes:

- Warrior
- Wizard
- Thief
- Necromancer

Prompt the user to select which class they want their character to be.

The minimum ability scores for each class are as follows:

Class	Strength	Intelligence	Wisdom	Dexterity	Constitution
Warrior	15	-	-	12	10
Wizard	-	15	10	10	-
Thief	10	9	-	15	-
Necromancer	10	10	15	-	-

If the character does not have any of the necessary minimum scores for the attributes required by the selected character class they can trade points from non-required attributes at the rate of 2 to 1 but cannot take an attribute below 3.

For example, a character is created with the following Base score and the player wants to be a warrior:

Class	Strength	Intelligence	Wisdom	Dexterity	Constitution
Base score	12	8	13	8	11
Warrior	15	-	-	12	10
Deficit	-3	-	-	-4	-
Surplus	-	5	10	-	-

Therefore the player has 5 in surplus for Intelligence of which 4 are usable (converting to 2 transferable points). The player also has 10 surplus points for Wisdom. 4 + 10 = 14 points to

exchange. They require 3 points for strength and 4 points on their dexterity; therefore (3 * 2) + (4 * 2) = 14; so they have enough points to be a warrior.

Your program should prompt the user to select which attributes they want to sacrifice, how many points they want to exchange and which attribute they want to assign them to.

In the example, the user takes 4 points from intelligence and assigns 2 to strength (strength: 14), then they take 2 from wisdom and assign to strength (strength: 15) finally they take 8 more from wisdom and assign to dexterity (dexterity: 12) and their final character attributes will be:

Class	Strength	Intelligence	Wisdom	Dexterity	Constitution
Amended score	15	4	3	12	11

If the player did not have sufficient point to exchange then the program should tell them they cannot be that character class and will have to start over again by re-running the program.

Try to solve this problem using Object Oriented Programming.