BIRZEIT UNIVERSITY

Engineering and Information Technology Faculty
Computer Science Department
Comp231-Advanced Programming

Programming Assignment #2

Due: Tuesday, 12/05/2022 (upload electronic copy by 11:59pm)

Objectives: (Chapter 1- Chapter 9).

Note: YOU CAN'T USE ANY OTHER CONCEPT OUTSIDE THIS CHAPTERS (WILL NEVER BE GRADED).

Write a Java program that does the following:

Your friends are trying to play a game with a set of teams to guess answers to some random mathematical functions. Some statistics about these operations are intake. Write a Java program that will calculate these statistics.

Input:

1. The user should be prompted to input the number of players N as an integer.
2. The user should be prompted to input information about the class **Player with the following attributes:** Name as String, age as integer, and height as double. All these data members are private.

Processing:

1. The program should divide those players (N players) **randomly** over the k-teams. Every team should have a minimum of $\frac{N}{K}$ -1 players and a maximum of $\frac{N}{K}$ +2 players.

Hint: *for the next iteration, you have to take care of the of remain players number from N players, and the number of teams. It will be decremented (i.e. N=N-Number of players for first team) and so on. The value of k is also decremented by 1 (i.e. k--).*

2. Store information (name, age, and height) about players for every team in a two-dimensional array. Rows represent the team number, and the number of columns represents the information about players. (as shown in the figure.1)

3. Call a method to find the average height of all players.

```
public static double avgHeight (Player [ ][ ] player) {..}
```

4. Call a method **choosePlayer**(`Player [ ][ ] player`){...}, that chooses two-player randomly from all of the teams. Those two-player must be selected from the different groups (not allowed to be from the same team). This method should return an array of object of Player type.

```
public static Player[ ] choosePlayer(Player [ ][ ] player ){.....}
```

Now, ready to start the competition between those players. Call a method **askPlayer**(Player[ ] player) that will ask those players **four random mathematical** questions, the player who answers more questions than the other will win, or sometimes they may get the same points. This method should return the result of this game to the main (i.e., the result for both players). In the main method, print on the screen the result. Consider these type of question for the players (*, /, %, $\lceil x \rceil$, $\lfloor x \rfloor$, x^y,$\sin x$, $\cos x$, $\log_{10} x$) . **Read two numbers randomly that are not necessary to be distinct from same category (same row)** that are stored in the two dimensional array (as shown in figure.2). Test bank numbers are stored in the array (initialized when declare the array) as the following: Index 0 for *, /, %, x^y, index 1 for $\lceil x \rceil$, $\lfloor x \rfloor$, index 2 for $\sin(x)$, $\cos(x)$ (x in degree so you have to covert to radian using method Math.toRadians(X), and index 3 for $\log_2 x$. User answers should be compared with a computer to consider the right answers.

```
public static int[ ] askPlayer(Player[ ] player ){.....}
```

6. Play this game until the user enter ('S'|| 's') to stop the game.
7. Print Information about the owner player.(Name,age,and height)

Sample output (not in details, this is for making an overview):

Please enter the number of players: 15

Please enter of teams: 3
Team #1 have: 5 players
Please enter information about players 1 into team 1 :
Name: Ali
Age:25
Height : 175
....
...

Team3	Player1	Player2	Player3	Player4	Player5	Player6
Team2	Player1	Player2	Player3	Player4		
Team1	Player1	Player2	Player3	Player4	Player5	

Figure.1

Team #2 have: 4 players
Please enter information about players 1 into team 2 :
Name: Sami
Age:25
Height : 185
....
...

2	4	8	16	32	64	128	256	512
30	45	60	90	180	270	135	180	0
1.8	2.5	-2.9	-1.5	1.3	1.4	2.9	3.6	4.5
2	3	1	5	9	8	4	7	6

Index 0 →

Figure.2

Team #3 have: 6 players
Please enter information about players 1 into team 3 :
Name: Najwa
Age:18
Height : 155
....
...

The average height of all players were : 168.5

Starting the Game: After Random selection from 3 teams, the two players we chose are:

1. **Player #1: from team #3 His/her name is: Haneen, with age 17, and his/her height is: 168**
2. **Player #2: from team #1 His/her name is: Ali, with age 25, and his/her height is: 175**

Steps for playing the game (you have to print the messages according to the following descriptions) :

* **Display Questions for player#1**

Question #1

Get Answer from Player (from a scanner) : if the answer is correct, increment the correct answers by 1 (no need to display any think)Question #2
Get Answer from Player : .....

* **Display Questions for player#2:**
Question #1
Get Answer from Player (from a scanner) : if the answer is correct, increment the correct answers by 1 (no need to display any think)
Question #2
Get Answer from Player : .....

The winner is Player#1
He/she has answered 4/4 question but player #2 has answered 3/4 questions

8. Enter 's' or 'S' to stop the game or another char to continue.

Additional Requirements:

1) Your program must compile and run, otherwise you will receive a score of 0.
2) Use appropriate data types. Use int unless you know the value could have non-zero digits after the decimal point.
3) Your program must output the correct values given any valid input values.

Style:

In particular:

1. Include the Header comments, like last time, including a good description
2. Variables: Use meaningful variable names and use camel case. Each variable declaration must be on a separate line with a descriptive comment.
3. Named constants: use these for numeric literals, and use uppercase and underscores in their names.
4. Source code lines should be less than 80 characters in length, and the program statements should be indented appropriately.

Specification Submission:

- Online through ITC.
- What to submit: Your own well-structured and well-commented JAVA files (.java)
- into a student Id_sec#.java file, e.g. sec1_120dddd.java).

Note:

*) **The questions and operations are chosen randomly, i.e., the input(s) and the operations:**
Hint: Operations can be stored in an array of character types ['m', 'p', 'i', 'f', 's', 'c', 'l'] where: m: Multiplication, p:power , i: ceil, f:floor, s: sin(x), c: cos(x), and l: log

Good Luck!!