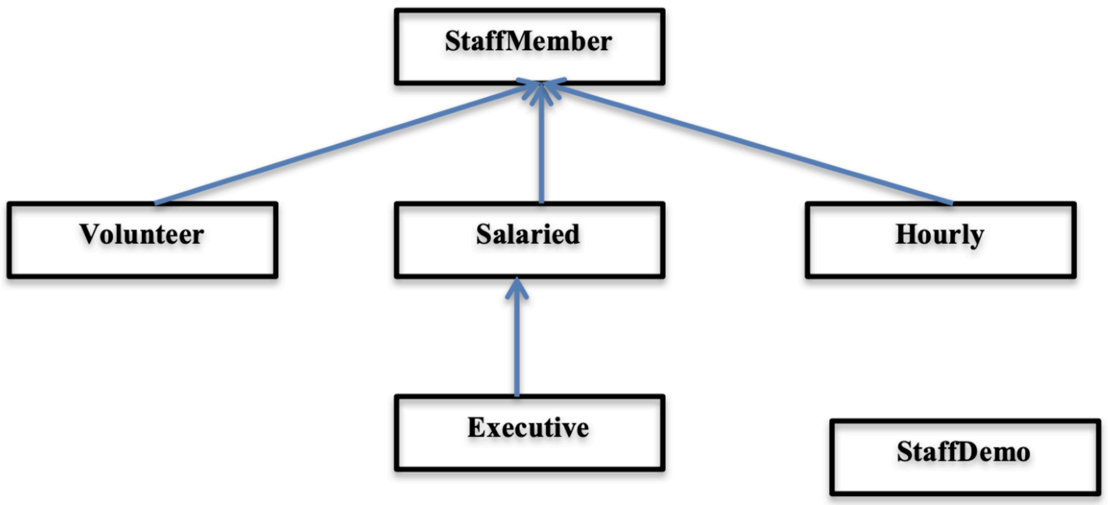
CSCI 218: Programming II (Spring 2025)

Midterm Examination

Due submission date: February 28th 2025, 11:59pm
This exam is worth a total of 20 points.

Question 1: The Employee Hierarchy (5 points)



Employees in our fictitious company are represented by the class hierarchy given above.

The StaffMember class is abstract, and is used to store common elements at appropriate levels. For example, StaffMembers all have first and last names, and StaffMember (Volunteer, Salaried, Hourly, and Executive) have an id number (int). They also have a pay rate (double).

The constructors of all staff member classes should accept parameters representing the id number, first name, last name, and pay rate (in that order). The pay rate can also be modified using a separate setter method called setPayRate().

All StaffMember objects can be paid, although this concept is undefined (abstract) at the StaffMember level.

Specific staff members are represented by Volunteer, Salaried, Hourly, and Executive objects. Volunteers receive no monetary compensation. All other StaffMember objects should manage a financial pay rate, which is interpreted differently depending on the type of employee. Assume that all employees are paid once per month.

The payment for Salaried employees should be interpreted as their annual salary. So the **pay** method should return their salary divided by 12. An Executive gets paid their salary plus any bonuses accumulated. A bonus is a one-shot deal – once paid, the bonus value goes back to zero. The bonus amount for each executive is 1% of his/her annual salary. The Executives may also accrue additional bonus amounts in a given month.

The payment for an Hourly employee should be interpreted as their per-hour pay rate. Hours are accumulated and payment is made accordingly. Once paid, accumulated hours go back to zero. The company policy is that the Hourly employees work for a minimum of 80 hours a month. These employees can also choose to work overtime and accrue more hours.

Appropriate functionality must be added to various classes to represent their processing. For example, the Executive employee will need an addBonus() method that accepts a double representing a bonus to be added to their current bonus (they could get more than one bonus in a given month). Hourly employees will need an addHours() method that accepts a double value to add to the accumulating hours worked.

Feel free to augment the definition of these classes as you see fit with other data and methods, but they are not intended to be realistic representations of employees in a real company.

Create a class called StaffDemo that includes the main() method to test the various functionality.

Question 2: Guess the Number Game (5 points)

Write a GUI application that plays "Guess the Number." It begins by choosing the number to be guessed by selecting an integer at random in the range 1-1000. The user will have ten tries to guess the number. When the game is over, the user should not be allowed to enter any more guesses. They should also have the option of starting the game over with a new random number to guess.

The game will be played using the following GUI components:

- A JLabel should display a message similar to, "I'm thinking of a number between 1 and 1000: Guess it!"
- A JTextField should be used to input the guess.
- A JLabel should tell the user how many guesses are remaining
- A JLabel should display either "Too High", "Too Low", "Correct!" or "You Lose!"

Question 3: Soccer Tournament (One- and Two-Dimensional Arrays) (5 points)

This program is about generating a road map of matches and related goal statistics in the FIFA World Cup Final. The game is repeated until your favorite soccer team (country) wins the Final in a tournament, or until a maximum number of tournaments. For simplicity, we will start from the Round of 16. Assume that you are given the list of 16 countries in an array. You will ask the user for his/her favorite team (keyboard input). Your program will then output a possible road map of 8 matches in the *Round of 16, 4 Quarter Finals, 2 Semi-Finals, 1 Final*, and the final winner of a tournament. A tournament is repeated (with the same list of countries) until the user's chosen team wins, or the max number of tournaments specified in the program has reached (e.g., 20 tournaments). At the end, you will show total scores in each match, the average goals for each tournament, the overall average for all tournaments in the game, and the total number of matches where the number of goals is greater than the overall average.

Note that a **match** is played between two teams/countries, a **tournament** consists of all the matches leading to a final winner (i.e., all matches including the final), and **the game** includes all the tournaments until your favorite team wins, or the max tournament count is reached.

Your program should proceed as follows:

1. The names of the teams are provided as an array of Strings.

```
String [] teams16 = {"Uruguay", "Portugal", "France", "Argentina",
"Brazil", "Mexico", "Belgium", "Japan", "Spain", "Russia", "Croatia",
"Denmark", "Sweden", "Switzerland", "Colombia", "England"};
```

2. Ask the user for her favorite team. If the provided team does not exist in the 16 names, you exit with a proper message (e.g., '`Your team is not in the Round of 16`'). Ignore the case, and any additional white spaces entered by the user.

3. Generate the output roadmap as follows. Choose teams in pairs for each match (e.g., take the first two teams from your array for Match 1, and the next two teams for Match 2, and so on). The winner of each match is selected as follows: in the 90-minute period, each team scores between 0 to 4 goals (generated randomly using the Java Random class), the team with more goals wins. If the play is a draw after 90 minutes (e.g., 2-2), there is a 30-minute *sudden death* period; assume that one team will score a goal during this period (chosen randomly), and that team will win (the play ends with the sudden death goal). In either case, you show the final score line, and the winner moves to the next phase. The final match winner is the FIFA World Cup Winner of 2026. Iterate the tournament until your favorite team wins (due to random goal selection, the score-lines will change each time), or you reach the maximum tournament count.

Sample output from the program (the user input is shown in grey):

```
Enter your favorite team: Japan
ROUND OF 16[Uruguay 1:2 Portugal] [France 0:1 Argentina] [Brazil 1:2
Mexico] [Belgium 1:3 Japan] [Spain 3:4 Russia] [Croatia 1:4 Denmark]
[Sweden 4:5 Switzerland] [Colombia 1:3 England]
QUARTER-FINALS [Portugal 4:1 Argentina] [Mexico 4:1 Japan] [Russia
0:1 Denmark] [Switzerland 4:1 England]
SEMI-FINALS [Portugal 3:4 Mexico] [Denmark 1:3 Switzerland]
FINAL [Mexico 3:1 Switzerland]
Tournament: 0 The WINNER is: Mexico


ROUND OF 16[Uruguay 0:1 Portugal] [France 2:1 Argentina] [Brazil 1:4
Mexico] [Belgium 2:4 Japan] [Spain 3:0 Russia] [Croatia 1:2 Denmark]
[Sweden 1:4 Switzerland] [Colombia 3:1
England]
QUARTER-FINALS[Portugal 2:1 France] [Mexico 0:2 Japan] [Spain 4:0
Denmark] [Switzerland 0:2 Colombia]
SEMI-FINALS[Portugal 4:1 Japan] [Spain 3:1 Colombia]
FINAL[Portugal 3:0 Spain]
Tournament: 1 The WINNER is: Portugal

ROUND OF 16[Uruguay 3:2 Portugal] [France 4:3 Argentina] [Brazil 4:0
Mexico] [Belgium 4:3 Japan] [Spain 4:3 Russia] [Croatia 4:5 Denmark]
[Sweden 0:4 Switzerland] [Colombia 0:1 England]
QUARTER-FINALS[Uruguay 2:3 France] [Brazil 4:1 Belgium] [Spain 0:2
Denmark] [Switzerland 1:0 England]
SEMI-FINALS[France 1:2 Brazil] [Denmark 2:0 Switzerland]
FINAL[Brazil 0:1 Denmark]
Tournament: 2 The WINNER is: Denmark

ROUND OF 16[Uruguay 2:3 Portugal] [France 0:3 Argentina] [Brazil 3:4
Mexico] [Belgium 1:4 Japan] [Spain 5:4 Russia] [Croatia 1:2 Denmark]
[Sweden 3:1 Switzerland] [Colombia 0:3 England]
QUARTER-FINALS[Portugal 3:1 Argentina] [Mexico 2:4 Japan] [Spain 3:4
Denmark] [Sweden 4:1 England]
SEMI-FINALS[Portugal 3:4 Japan] [Denmark 4:5 Sweden]
FINAL[Japan 1:0 Sweden]
Tournament: 3 The WINNER is: Japan
```

4. You also need to keep counts of the total goals scored in each play, and at the end of the game (either when your team won, or the max tournament count has reached), you show the following: total goals from each match in a tournament and the average goal for each tournament; the average goal for the entire game (all matches in all tournaments); the number of matches with scores greater than the overall average. After each match, you need to store the total goals in that match for calculating the above statistics; a 2D array will be needed. Note that there are 15 matches in a tournament. A single digit is kept after the decimal point for the real numbers (averages). Sample output (following the score lines in the part 3 example):

```
GOAL STATS
[Tournament 0] Total goals: [3, 1, 3, 4, 7, 5, 9, 4, 5, 5, 1, 5, 7, 4, 4] [Average: 4.5]
[Tournament 1] Total goals: [1, 3, 5, 6, 3, 3, 5, 4, 3, 2, 4, 2, 5, 4, 3] [Average: 3.5]
[Tournament 2] Total goals: [5, 7, 4, 7, 7, 9, 4, 1, 5, 5, 2, 1, 3, 2, 1] [Average: 4.2]
[Tournament 3] Total goals: [5, 3, 7, 5, 9, 3, 4, 3, 4, 6, 7, 5, 7, 9, 1] [Average: 5.2]
Average goals for 4 tournament(s): 4.3
Total matches in all tournaments over the average goal value: 27
```

5. If your team does not win (i.e., the game ended when you reach the max tournament count), show the following message after the output in part 3 above (instead of showing ``It took 4 tournament(s) of the game for Japan to win!!!'').

```
Sorry, Japan didn't win in 20 tournaments!
```

You still need to show the goal statistics as in part 4.

Question 4: Code Report and Evaluation (5 points)

- Submit the source code files **on Moodle**.
- Ensure that you follow proper Object-Oriented Programming (OOP) practices throughout your code.
- Include a screen recording demonstrating the testing of all your code implementation.
- Provide a concise report explaining your approach to solving the tasks, the design decisions you made, and any challenges you faced during implementation.