PROG-2200 Assignment 1

Title

Review of Java Concepts

Value

10%

Learning Outcomes Evaluated

Develop an application utilizing custom-developed or third-party class libraries or frameworks.

Instructions

Objective:

Create a command-line program that allows users to manage a hockey team, including adding players and coaches, viewing the team roster, and tracking player statistics (suggestion: ice time etc.) The program should include a class and at least two subclasses (Person, Player, Coach), handle user input (scanner) with try/catch exceptions, use enums (for example, player position), arrays, and produce output.

For example, the program should prompt the user with the following:

- Enter a name for your team:
    - Enter name:
    - Choose an option: 1. Add Player 2. Add Coach 3. View Team 4. Remove Member 5. Exit
- When adding a player:
    - Enter player name:
    - Enter player age:
    - Enter player position:
    - Enter player jersey number:
    - Enter ice time (or other statistics)
- When adding a coach:
    - Enter coach name:
    - Enter coach age:
    - Enter coach role:
- When Viewing the Team (output example):
    - Team: NSCC Hawks (HOCKEY)
    - Player: Jamie, Age: 20, Position: Forward, Jersey Number: 12, Ice Time: 123
    - Coach: Fred, Age: 40, Role: Head Coach
- When removing a player:
    - Enter the ~~name~~ jersey number of the member to remove:

Requirements:

Class Structure:

- Base Class: Person
- Subclasses: Player, Coach
- Additional Class: Team

Enum:

- Create an enum called Position with values, such as: Center, Goalie, Right-Wing, Left-Wing, Defense, Forward.

User Input and Exception Handling:

- Prompt the user to add, view, or remove players and coaches.
- Handle invalid input with try/catch blocks.

Arrays:

- Use an array to store the team members.
- ~~Use an array to manage user commands.~~

Output:

- Display Team info - player and coach details.
- ~~Display team rosters and statistics.~~

Deliverables

ZIP file or GitHub URL

Evaluation

Learning Outcomes	Components	Points
Develop an application utilizing custom-developed or third-party class libraries or frameworks.	class, and at least two subclasses	2
	handle user input (and try/catch exceptions)	4
	require an enum	2
	include arrays	2
	output	2
Outcome Mark Value		/12

Grading Rubric

Base Class and Subclasses (2 points)

- 2 points: The program includes a well-defined base class (Person) and at least two subclasses (Player and Coach) with appropriate inheritance and methods.
- 1 point: The program includes a base class and only one subclass, or the subclasses lack proper methods and properties.
- 0 points: The program does not include a base class and/or subclasses.

Handling User Input and Try/Catch Exceptions (4 points)

- 4 points: The program effectively handles user input for all required actions (adding, viewing, removing members) and uses try/catch blocks to manage invalid inputs, providing appropriate error messages.
- 3 points: The program handles most user inputs correctly and includes try/catch blocks, but some cases are not covered or error messages are unclear.
- 2 points: The program handles user input but lacks proper use of try/catch blocks, or try/catch blocks are present but not effectively used.
- 1 point: The program attempts to handle user input but fails to include try/catch blocks, leading to frequent unhandled exceptions.
- 0 points: The program does not handle user input or include any try/catch exceptions.

Enum Requirement (2 points)

- 2 points: The program includes an enum (Position) that is used appropriately within the program
  1 point: The program includes an enum but does not use it effectively or appropriately within the program.
- 0 points: The program does not include an enum.

Include Arrays (2 points)

- 2 points: The program effectively uses arrays (or ArrayLists) to store and manage team members and other relevant data.
- 1 point: The program uses arrays but not effectively, leading to issues in managing or displaying data.
- 0 points: The program does not include arrays.

Output (2 points)

- 2 points: The program produces clear and correct output for all required actions (displaying team rosters, player and coach details).
- 1 point: The program produces output but it is unclear, incomplete, or occasionally incorrect.
- 0 points: The program does not produce any output or the output is consistently incorrect.

Total: 12 points