**Team Management System Using Python Programming**

**Project Assignment**

**GIK29B Python- and R-programming**

**Prepared By: Dhiraj Gyawali**

**Date Prepared: 2024-03-13.**

Contents

[Introduction 2](#_Toc161250402)

[Description 2](#_Toc161250403)

[ Team Class 2](#_Toc161250404)

[ Function Class 2](#_Toc161250405)

[ UserInterface Class 3](#_Toc161250406)

[ Main Program 3](#_Toc161250407)

[User Guide 3](#_Toc161250408)

[Overview**:** 3](#_Toc161250409)

[Getting Started: 3](#_Toc161250410)

[Running the Program: 3](#_Toc161250411)

[Program Features: 4](#_Toc161250412)

[Using the Program: 4](#_Toc161250413)

[Tips: 4](#_Toc161250414)

[Exiting the Program: 4](#_Toc161250415)

[Support: 4](#_Toc161250416)

# Introduction

This project is the python assignment project where the task is to create a working team management system for different hockey teams participating in youth hockey cup. A menu interface-based approach is implemented where the user can simply select the menu and perform operations. The main objective of this program is to keep records of all the teams participating in the cup. Here the user can easily enter the team details like team name, type of team either boy or girl, whether the fee have been paid for participation or not and the amount required for registration. The team id and the date the team was created are automatically set when the user enters other details about the team. Similarly, the user can display the team details based on team id or team type, modify the team details after creating the team and delete any team if they want to remove it. More functionalities are added by introducing more menu options. Users can display team statistics like total number of teams and percentage who has paid their fees. The user can also cancel a team participation by adding a date in the team information if the team wants to remove their participation. Saving the team information to a .txt file and later restoring that .txt file to load information about teams is another feature.

# Description

We have created three classes here and one main function to run the program. Let’s discuss in detail each class and their solution.

* Team Class**:** This class consists of private instances related with the teams like id, date created, name, team type, fee paid status, amount of fees required for their registration and cancellation date if the team had any. We have used a counter to create the id of the teams. Datetime module was used to create the date and time automatically when the team was created. Name, type of team and feed paid status was to be entered by user. The fee amount required for registration will be set by the user and will enter it. Then we define set functions for all the instances except date and time to set the instances to certain value user enter. The get method was also defined for every instance so it can be called whenever we want to access the values of those instances. This class has object\_team as the object of class to keep record of each team which is used in the Function class to perform operations.
* Function Class**:** We have defined a number of functions in this class to perform operations based on the user’s need. The first function was create\_team() function which helps user to enter the team details. The second function was read\_team() which displays the team information based on the user requirement. User can select to read all teams, to read based on boy or girl team, or just by entering the id of the team they want to read. Third function was update\_team() which updates the content of the team based on the user need. The user can enter the id of the team they want to update and can choose what instance they want to update and update it accordingly. The fourth function was delete\_team () function which helps user to delete any team they want to remove from the team information list. Fifth function was team\_statistics() which display the total number of teams and percentage of teams that have paid their fees. Sixth function is cancel\_team\_participation() function which allows user to mark any team as cancelled by adding the cancelled date to its information. Another function is save\_teams\_to\_file() and restore\_teams\_from\_file() which will allow the user to save the team information in a text file and later can reload that text file to perform operations.
* UserInterface Class**:** This class defines all the menu based interface system that a user can easily utilize to work on the teams information. A display\_menu() function is defined in this class which asks the user their inputs based on the different menu options created and performs menu specific tasks. This class uses the functions from Function class to perform operations.
* Main Program**:** This program is used to run our system that we have created using all these classes and functions. It uses UserInterface class to load the menu and perform operations.

# User Guide

## Overview**:**

This Team Management System is a Python program designed to help manage information about hockey teams participating in youth hockey cup. It allows users to create, view, update, and delete team recordswith other added functionalities like displaying team statistics,cancelling team, save the information to a text file and reload the text file whenever necessary.

## Getting Started:

* To use the program, ensure you have Python installed on your computer.
* Download the program files from the repository or source.
* Open the command line or terminal.
* Navigate to the directory where the program files are located.

## Running the Program:

Run main.py program using the Python interpreter.

## Program Features:

* Create Team: Allows users to create a new team record by providing information such as team name, type, payment status, and fee amount.
* Update Team: Enables users to update existing team records, including changing the team name, type, payment status, or fee amount.
* Delete Team: Allows users to delete a team record from the system.
* View Team: Provides options to view team records based on different criteria, such as viewing all teams, viewing by team ID, or filtering by team type.
* Team Statistics: Allows users to view total number of teams and percentage of teams that have paid the fees.
* Cancel Team: Allows user to keep track of the teams that want to cancel their participation by adding a cancelled date.
* Save Teams to File: Saves the current list of teams to a text file for future reference.
* Restore Teams from File: Restores team records from a previously saved text file.

## Using the Program:

* Follow the on-screen prompts to navigate through the program's menu options.
* Input data as requested and follow the instructions to perform various actions such as creating, updating, or viewing team records.
* Use the save and restore options to store team information persistently and reload it later.

## Tips:

* Ensure to provide valid inputs when prompted to avoid errors.
* Use meaningful names for teams to easily identify them.
* Regularly save team information to a file to prevent data loss.

## Exiting the Program:

To exit the program, choose the appropriate option from the menu or close the terminal/command prompt window.

## Support:

For any issues or feedback, contact the program developer.