

# CS232 Lab - Vector of Objects

## Create a program that stores baseball player information into a vector

### User Requirements

Create a menu system that allows the user to enter, remove and modify a team of baseball players as seen in the figure below (see Figure 1: Sample program execution).

```
1 - add a player to the team
2 - remove a player from the team
3 - change a player's jersey number
4 - change a player's availability
5 - change a player's name
6 - display team
7 - quit
=====
Enter your choice: _
```

```
Enter your choice: 1
What is the player's name: Robert
What is Robert's jersey number: 7
Is Robert available to play (y/n): y
```

```
Enter your choice: 3
Enter the name of the player who's jersey you want to change: Robert
What is Robert's new jersey number: 99
Robert's jersey number is now 99
```

```
Enter your choice: 5
Enter the name of the player who's name you want to change: Robert
What is the updated replacement name: Bob
Robert's name has now been updated to Bob
```

```
Enter your choice: 2
What is the name of the player you want to remove: Robert
Robert's record has been removed
```

```
Enter your choice: 4
Enter the name of the player who's availability you want to change: R
Is Robert available to play (y/n): n
Robert's status is now set to not available
```

```
Enter your choice: 6
Name    Number  Availale
Joshua  44       Yes
Jackson 27       Yes
```

Figure 1: Sample program execution

### Software Requirements

- Create a vector of baseball player objects that gives the user the ability to enter a new player, remove any given player, change the players jersey number, change the availability of a player, change the name of a player and finally display the team roster (see Figure 1).
- Create a class called BaseballPlayers as seen in the header file image below

```
#include<string>

class BaseballPlayers {
public:
    BaseballPlayers(); //default constructor
    BaseballPlayers(int jerseyNumPar, bool isAvailablePar, std::string namePar);
    void setJerseyNumber(int jerseyNumPar);
    void setIsAvailable(bool isAvailablePar);
    void setFirstName(std::string namePar);

    int getJerseyNumber();
    bool getIsAvailable();
    std::string getName();

    void changeJersey(int newJerseyNumPar, std::string firstNamePar);
    void changeName(std::string newPlayerNamePar, std::string firstNamePar);
    void changeAvailability(bool newAvailability, std::string firstNamePar);
private:
    int jerseyNumber;
    bool isAvailable;
    std::string firstName; //note, string lives in namespace std
};
```

Figure 2: BaseballPlayers Header File

## CS232 Lab - Vector of Objects

- Here are the specifics for the BaseballPlayers class members
  - BaseballPlayers() – this default constructor should assign jerseyNumber to 0, isAvailable to false and firstName to "" (i.e., blank).
  - BaseballPlayers(int jerseyNumPar, bool isAvailablePar, std::string namePar) – the overloaded constructor shall assign the arguments with the values provided by the passed parameters.
  - 
  - void setJerseyNumber(int jerseyNumPar) – assigns jerseyNumPar to the private instance variable jerseyNumber
  - void setIsAvailable(bool isAvailablePar) – assigns isAvailablePar to the private instance variable isAvailable
  - void setFirstName(std::string namePar) – assigns namePar to the private instance variable firstName
  - 
  - int getJerseyNumber() – returns the value of the private instance variable jerseyNumber
  - bool getIsAvailable() – returns the value of the private instance variable isAvailable
  - string getName() – returns the value of the private instance variable firstName
  - 
  - void changeJersey(int newJerseyNumPar, std::string firstNamePar) – the firstNamePar is compared to the firstName private instance variable. If they are the same the call the setJerseyNumber class function to assign newJerseyNumPar to jerseyNumber.
  - void changeName(std::string newPlayerNamePar, std::string firstNamePar) – the firstNamePar is compared to the firstName private instance variable. If they are the same then call the setFirstName class function to assign newPlayerNamePar to firstName.
  - void changeAvailability(bool newAvailability, std::string firstNamePar)– the firstNamePar is compared to the firstName private instance variable. If they are the same then call the setIsAvailable class function to assign newAvailability to isAvailable.