Myarray project player

```
#include"PlayerArr.h"


/* run this program using the console pauser or add your own getch, system("pause") or input loop
*/


int main() {
    PlayerArr Ply(10);
    Ply.displayPly();
    int choice;
    do{

    cout<<"\n\n===========================================================\n\n";
        cout<<"========Welcome to the Player management sytem=============\n";
        cout<<"0.Exit\n";
        cout<<"1.AddPlayer:\n";
        cout<<"2.Search Player:\n";
        cout<<"3.Delete Player:\n";
        cout<<"4.Update Player:\n";
        cout<<"5.Sort Players;\n";
        cout<<"6.Display All Player:\n";

        cout<<"Enter the choice:\n";
        cin>>choice;

        switch(choice){
            case 0:{
                cout<<"Thanks for visiting!!\n";
                break;
            }
            case 1:{
                int jn;
```

```
		char pname[20];
		int run;
		int wickets;
		int matches;
		cout<<"\nEnter the details of Players:\n";
		cout<<"Enter the jersy number:\n";
		cin>>jn;
		cout<<"Enter the name of player:\n";
		cin>>pname;
		cout<<"Enter the number of runs:\n";
		cin>>run;
		cout<<"Enter the number of wickets:\n";
		cin>>wickets;
		cout<<"Enter the number of matches:\n";
		cin>>matches;

		Player p(jn,pname,run,wickets,matches);
		Ply.addPly(p);
		break;
	}
	case 2:{
		int ch;
		cout<<"Enter the choice:";
		cin>>ch;
		switch(ch){
			case 1:{
					int jn;
					cout<<"Enter the jersy number:\n";
					cin>>jn;
					int i=Ply.searchPlyJn(jn);
					Ply.getPtr()[i].display();
```

```
            break;
        }
        case 2:{
            char name[20];
            cout<<"Enter the name:";
            cin>>name;
            int i=Ply.searchPlyName(name);
            Ply.getPtr()[i].display();
            break;
        }
        case 0:{
            cout<<"Exit";
            break;
        }
        default:{
            cout<<"Invalid Input\n";
            break;
        }
    }

    break;
}
case 3:{
    int jn;
    cout<<"Enter the jersy number:\n";
    cin>>jn;
    Ply.deletePly(jn);

    break;
}
case 4:{
```

```
                int jn;
                cout<<"Enter the jersy number:\n";
                cin>>jn;
                Ply.updatePly(jn);
                break;
            }
            case 5:{
                Ply.sort();
                break;
            }
            case 6:
            {
                Ply.displayPly();
                break;
            }
        }

    }while(choice!=0);
    return 0;
}
#include<iostream>
using namespace std;
class Player{
    int jersyNo;
    char name[20];
    int noRuns;
    int noWickets;
    int noMatches;

    public:
        Player();
        Player(int ,char*,int,int ,int);
        void setjersyNo(int);
        void setName(char*);
        void setNoRuns(int);
        void setNoWickets(int);
        void setNoMatches(int);
        int getJersyNum();
        char* getName();
        int getRuns();
        int getwickets();
        int getNoMatches();
        void display();

};
#include"player.h"

Player::Player(){
    this->jersyNo=0;
    strcpy(this->name,"player");
    this->noRuns=0;
    this->noWickets=0;
    this->noMatches=0;
}
Player::Player(int jn ,char* nm,int run,int  wic,int mat){
    this->jersyNo=jn;
    strcpy(this->name,nm);
    this->noRuns=run;
    this->noWickets=wic;
    this->noMatches=mat;
}
void Player::setjersyNo(int jn){
```

```
                this->jersyNo=jn;
}
void Player::setName(char* nm){
                strcpy(this->name,nm);
}
void Player::setNoRuns(int run){
                this->noRuns=run;
}
void Player::setNoWickets(int wic){
                this->noWickets=wic;
}
void Player::setNoMatches(int mat){
                this->noMatches=mat;
}

int Player::getJersyNum(){
        return this->jersyNo;
}
char* Player::getName(){
        return this->name;
}
int Player::getRuns(){
        return this->noRuns;
}
int Player::getwickets(){
        return this->noWickets;
}
int Player::getNoMatches(){
        return this->noMatches;
}
void Player::display(){
	cout<<"\nPlayer\n";
	cout<<"Player's jersyNo: "<<this->jersyNo<<"\n";
	cout<<"Player's name:"<<this->name<<"\n";
	cout<<"Player's noRuns:"<<this->noRuns<<"\n";
	cout<<"Player's noWickets:"<<this->noWickets<<"\n";
	cout<<"Player's noMatches:"<<this->noMatches<<"\n";
}
#include "player.h"
class PlayerArr{
	int size;
	int index;
	Player* ptr;

	public:

	Player* getPtr();
	PlayerArr(PlayerArr&);//copy constructor
	PlayerArr(int);
	bool isFull();
	bool isEmpty();
	bool addPly(Player&);
	int searchPlyJn(int);
	int searchPlyName(char*);
	void deletePly(int);
	void updatePly(int);
	void displayPly();
	void sort();
};
#include"PlayerArr.h"
PlayerArr::PlayerArr(int s){
	this->size=s;
	this->index=-1;
	this->ptr= new Player[size];
	//hardcoded values
	this->ptr[0].setjersyNo(101);
	this->ptr[0].setName("Prachiti");
	this->ptr[0].setNoRuns(100);
	this->ptr[0].setNoWickets(10);
	this->ptr[0].setNoMatches(1);

	index++;

	this->ptr[1].setjersyNo(102);
	this->ptr[1].setName("Hrutuja");
	this->ptr[1].setNoRuns(444);
	this->ptr[1].setNoWickets(44);
	this->ptr[1].setNoMatches(4);

	index++;

	this->ptr[2].setjersyNo(103);
	this->ptr[2].setName("Shreya");
	this->ptr[2].setNoRuns(555);
	this->ptr[2].setNoWickets(55);
	this->ptr[2].setNoMatches(5);

	index++;


}
bool PlayerArr::isFull(){
	if(this->index==(this->size -1)){
```

```
                return true;
        }
        else{
                return false;
        }
}
bool PlayerArr::isEmpty(){
        if(this->index==-1){
                return true;
        }
        else
        {
                return false;
        }
}
bool PlayerArr::addPly(Player& p){
        //check arry is full
        if(isFull()){
                cout<<"Array is fulled!\n";
                return false;

        }
        else{
                index++;
                this->ptr[index]=p;
                return true;
        }
}

int PlayerArr::searchPlyJn(int jn){
        if(isEmpty()){
```

```
        cout<<"Arry is empty!!";
        return -1;
    }else{
        for(int i=0;i<=index;i++){
            if(jn==ptr[i].getJersyNum()){
                return i;
            }
        }//out of for

        return -1;
    }
}
int PlayerArr::searchPlyName(char* nm){
    if(isEmpty()){
        cout<<"Arry is empty!!";
        return -1;
    }
    else{
        for(int i=0;i<=index;i++){
                    if(strcmp(this->getPtr()[i].getName(),nm)==0){
                    return i;
                }
        }
        return -1;

    }
}
void PlayerArr::deletePly(int jn){
    if(isEmpty()){
        cout<<"Array is Empty!!";
    }
    else{

        int pos=searchPlyJn(jn);
        if(pos!=-1){
                if(pos==size-1){
                    index--;
                    }
                else{
                    for(int i=pos;i<index;i++){
                        ptr[i]=ptr[i+1];
                        }

                    index--;
                    }

        }
        else{
            cout<<"\nElement not found\n";
            }
        }
    }

void PlayerArr::updatePly(int jn){
    if(isEmpty()){
        cout<<"array is Empty!!\n";
    }
    else{
        int pos=searchPlyJn(jn);
        if(pos!=-1){
            int ch;
```

```
cout<<"1.Runs\n";
cout<<"2.wickets\n";
cout<<"3.Matches\n";
cout<<"Enter the choice:\n";
cin>>ch;
switch(ch){
        case 1:{
                int runs;
                cout<<"Enter the updated runs:\n";
                cin>>runs;
                ptr[pos].setNoRuns(runs);
                ptr[pos].display();
                break;
        }
        case 2:{
                int Wickets;
                cout<<"Enter the updated wickets:\n";
                cin>>Wickets;
                ptr[pos].setNoWickets(Wickets);
                ptr[pos].display();

                break;
        }
        case 3:{
                int matches;
                cout<<"Enter the updated matches:";
                cin>>matches;
                ptr[pos].setNoMatches(matches);
                ptr[pos].display();
                break;
        }
```

```
            }
        }
        else{
            cout<<"Element not found!!\n";
        }

    }
}
//sort
void PlayerArr::displayPly(){
    for(int i=0;i<=index;i++){
        ptr[i].display();
        cout<<"\n";
    }
}

Player* PlayerArr::getPtr(){
    return this->ptr;
}
PlayerArr :: PlayerArr(PlayerArr& ply){
    this->size=ply.size;
    this->index=ply.index;
    this->ptr=new Player[size];
    for(int i=0;i<=index;i++){
    this->ptr[i]=ply.ptr[i];
    }
}

void PlayerArr::sort(){
    PlayerArr p(*this);//call the copy constructor
    //now p is our copy of obj now perform the sorting on this p's arry
```

```
//sort by run .wickets.matches
int srt;
cout<<"1.Ascending\n";
cout<<"2.Descending\n";
cout<<"0.Exit\n";
cout<<"Enter the choice:\n";
cin>>srt;
switch(srt){
    case 1:{
                    int choice;
                    cout<<"Ascending\n";
                    cout<<"0.Exit\n";
                    cout<<"1.Run\n";
                    cout<<"2.Wickets\n";

                    cout<<"Enter the choice:";
                    cin>>choice;
                    switch(choice){
                        case 0:{
                            cout<<"Exit\n";
                            break;
                        }
                        case 1:{
                            //sort by runs
                            int min;
                            int pos=0;
                            for(int i=0;i<=p.index;i++){
                                min = p.ptr[i].getRuns();
                                pos=i;
```

```
                                        for(int
j=i+1;j<=p.index;j++){

    if(p.ptr[j].getRuns()<min){
                                            pos=j;
                                        }
                                    }
                                    Player temp =p.ptr[pos];
                                    p.ptr[pos]=p.ptr[i];
                                    p.ptr[i]=temp;
                                    //swap the element
                                }

                                //display all players after sorting
                                cout<<"ascending by Runs:";
                                p.displayPly();

                                break;
                            }
                            case 2:{

                                int min;
                                int pos=0;
                                for(int i=0;i<=p.index;i++){
                                    min = p.ptr[i].getwickets();
                                    pos=i;
                                        for(int
j=i+1;j<=p.index;j++){


    if(p.ptr[j].getwickets()<min){

    pos=j;
```

```
                                                            }
                                            }
                                            Player temp=p.ptr[pos];
                                            p.ptr[pos]=p.ptr[i];
                                            p.ptr[i]=temp;
                                            //swap the element
                                    }
                                    //display all players after sorting
                                    cout<<"ascending by Wickets:";
                                    p.displayPly();



                                    break;
                            }

                            default:{
                                    cout<<"Invalid inputs\n";
                                    break;
                            }

                    }
        break;
}
case 2:{
                    int choice;
                    cout<<"descending\n";
                    cout<<"0.Exit\n";
                    cout<<"1.Run\n";
                    cout<<"2.Wickets\n";

                    cout<<"Enter the choice:";
```

```
                        cin>>choice;
                        switch(choice){
                            case 0:{
                                cout<<"Exit\n";
                                break;
                            }
                            case 1:{
                                //sort by runs
                                int max;
                                int pos=0;
                                for(int i=0;i<=p.index;i++){
                                    max = p.ptr[i].getRuns();
                                    pos=i;
                                        for(int
j=i+1;j<=p.index;j++){

    if(p.ptr[j].getRuns()>max){
                                            pos=j;
                                        }
                                    }
                                    Player temp =p.ptr[pos];
                                    p.ptr[pos]=p.ptr[i];
                                    p.ptr[i]=temp;
                                    //swap the element
                                }

                                //display all players after sorting
                                cout<<"Descending by runs:";
                                p.displayPly();

                                break;
                            }
                            case 2:{

                                int max;
                                int pos=0;
                                for(int i=0;i<=p.index;i++){
                                    max = p.ptr[i].getwickets();
                                    pos=i;
                                        for(int
j=i+1;j<=p.index;j++){

    if(p.ptr[j].getwickets()>max){
                                            pos=j;
                                        }
                                    }
                                    Player temp=p.ptr[pos];
                                    p.ptr[pos]=p.ptr[i];
                                    p.ptr[i]=temp;
                                    //swap the element
                                }
                                //display all players after sorting
                                cout<<"Descending by Wickets:";
                                p.displayPly();


                                break;
                            }

                            default:{
                                cout<<"Invalid inputs\n";
                                break;
                            }
```

```
                                                  }
                              break;
                        }
                  }

            }
```