

```
#include
<iostream>
```

```
#include <conio.h>
#include <string.h>
#include <math.h>
#include <stdio.h>
```

```
using namespace std;
```

```
char* SHOW_STATUS(int i)
{
    switch (i)
    {
        case 0: return "";
        case 1: return "not out";
        case 2: return "bold out";
        case 3: return "caught out";
        case 4: return "run out";
        default: return "unknown";
    }
    return "unknown";
}
```

```
class Team
{
    char team_name[20];
    char p_name[12][20];
    int p_run[12];
    int p_status[12];
    int p_ball[12];
    int p_num;
    int Extra_run;
    public:

    Team();

    int Init(void);
    char* get_name(int i)
    {
        return p_name[i];
    }
    int get_run(int i)
```

```

        {
            return p_run[i];
        }
int get_status(int i)
{
    return p_status[i];
}
int get_extra(void)
{
    return Extra_run;
}
int get_ball(int i)
{
    return p_ball[i];
}
char* get_t_name()
{
    return team_name;
}
int get_Total(void);
void Add_Run(int player_num,int run);
void Set_Status(int player,int status)
{
    p_status[player]=status;
};
void Set_Out(int out_player,int type,int new_player);
void Set_Ball(int player)
{
    p_ball[player]++;
}
void Set_Extra(int run)
{
    Extra_run+=run;
};
};

```

/***** Default Contructor*****/

```

Team::Team()
{

```

```

        for(int i=1; i<=12;i++)
        {
            strcpy(p_name[i],"Player");
            p_run[i]=0;
            p_status[i]=0;
            p_ball[i]=0;
        }
        strcpy(team_name,"INDIA");
        Extra_run=0;
        p_num=0;
    }

```

/****** Initialize Team******/

```

int Team::Init(void)
{
    cout<<"Enter Team Name: ";
    cin>>team_name;
    cout<<"No of players of the team: ";
    cin>>p_num;
    cout<<"Enter Name of the Players.....\n";
    for(int i=1;i<=p_num;i++)
    {
        cout<<"                Player"<<i<<": ";
        cin>>p_name[i];
        cout<<"\n\n";
        p_run[i]=0;
        p_status[i]=0;
    }
    return p_num;
}

```

```

void Team::Add_Run(int player_num,int run)
{
    p_run[player_num] += run;
    get_Total();
}

```

```
/****** Calculate & Get Total Run *****/
```

```
int Team::get_Total()
{
    int local_total=0;
    for(int i=1;i<=p_num;i++)
    {
        local_total += p_run[i];           //Players Score
        local_total += Extra_run;          //Add Extra
    }
    return local_total;
}
```

```
void Team::Set_Out(int out_player,int type,int new_player)
{
    p_status[out_player]=type;             //Current Player out
    p_status[new_player]=1;                //New Player Not out
}
```

```
/******my score*****/
```

```
class myscore
{
    int player1;
    int player2;
    int Ball_count;
    int Over;
    int Max_Over;
    int Out;
    int Max_out;
    Team team1;
    int extra;
    public:
        myscore();
        void show(void);
        void ch(void);
        void dot_ball(void);
        void add_run(void);
}
```

```

        void wicket(void);
        void extra1(void);
        void over_complete(void);
};

myscore::myscore()
{
    player1=0;
    player2=0;
    extra=0;
    Over=0;
    Out=0;
    Max_out=10;
    Ball_count=0;
    Max_out=team1.Init(); // Initiali
    cout<<"\nEnter no of Overs in one Innings: ";
    cin>>Max_Over;
    cout<<"\n\nChoose opening Batsman(1-11):";
    while(1)
    {
        cout<<"\n                batsman 1:"; //Chose
        cin>>player1; //Opening
        team1.Set_Status(player1,1); //Batsman
        cout<<"                batsman 2:";
        cin>>player2;
        team1.Set_Status(player2,1);

        if(player1>11 || player2>11 || player1==player2)
            cout<<"Invalid Entry Try Again!!";
        else
            break;
    }
}

/***** Showing Entire Score *****/

void myscore::show(void)
{

```

```

cout<<"/////////////////////////////////////////"<<endl;
cout<<"                "<<team1.get_t_name()<<" Score
cout<<"/////////////////////////////////////////"<<endl;

for(int i=1;i<=Max_out;i++)
{
    cout<<team1.get_name(i)<<"                ";
    cout<<SHOW_STATUS(team1.get_status(i))<<"
    cout<<team1.get_run(i);
    if(team1.get_status(i))cout<<"("<<team1.get_ball(i)<<")";
    cout<<endl;
}

cout<<"\nExtra";
cout<<"\t"<<team1.get_extra()<<endl<<endl;
cout<<"\n\n-----"
cout<<"Over "<<Over<<". "<<Ball_count<<" Wicket "<<Out;
cout<<" || total score: ";
cout<<team1.get_Total()<<endl;

if(Ball_count<=6*Max_Over)
{
    ch();
}

}

void myscore::ch()
{
    cout<<"\nChoose option:\n";                //Options for
    cout<<"\n\tDot Ball[1]||";                //Updating
    cout<<"\n\tAdd Run[2]||";                //Score Card
    cout<<"\n\tExtra[3]||";
    cout<<"\n\tWicket[4]||";
    cout<<"\n\tExit[10]\n";
    cout<<"\t\t\t.....";
    int option;
    cin>>option;

    switch (option)

```

```

{
case 1:
            dot_ball();
            break;

case 2:
            add_run();
            break;

case 3:
            extra1();
            break;

case 4:
            wicket();
            break;

case 10:
            return;

default:
            cout<<"\nInvalid input\n";

            getch();
            dot_ball();
}
}

/***** One Dot Ball *****/

void myscore::dot_ball(void)
{
    cout<<"\n\n*****\n";
    cout<<"            Dot Ball            \n";
    cout<<"*****\n\n";
    Ball_count++;
    team1.Set_Ball(player1);

    if(Ball_count==6)
    {
        over_complete();
        return;
    }

    getch();
    show();
}

```

```
}
```

```
/****** Add Extra Run *****/
```

```
void myscore::extra1(void)
```

```
{
```

```
    cout<<"\n\n*****\n";
```

```
    cout<<"                Extra Run                \n";
```

```
    cout<<"*****\n\n";
```

```
    cout<<"Extra?";
```

```
    cin>>extra;
```

```
    team1.Set_Extra(extra);
```

```
    show();
```

```
}
```

```
/****** Add Current Player Run*****/
```

```
void myscore::add_run(void)
```

```
{
```

```
    cout<<"\n\n*****\n";
```

```
    cout<<"                Add Run                \n";
```

```
    cout<<"*****\n\n";
```

```
    cout<<"Runs? ";
```

```
    int runs;
```

```
    Ball_count++;
```

```
    team1.Set_Ball(player1);
```

```
    cin>>runs;
```

```
    team1.Add_Run(player1,runs);
```

```
    if(runs==1||runs==3)
```

```
    {
```

```
        int temp = player1;
```

```
        player1=player2;
```

```
        player2=temp;
```

```
    }
```

```
    if(Ball_count==6)
```

```
    {
```



```

        over_complete();
        return;
    }

    show();
}

/***** Over Complete *****/

```

```

void myscore::over_complete(void)
{
    cout<<"\n\n*****\n";
    cout<<"          Over Complete\n";
    cout<<"*****\n";
    Over++;
    Ball_count=0;
    int temp = player1;
    player1=player2;
    player2=temp;

    if(Over==Max_Over)
    {
        cout<<"\n\n*****\n";
        cout<<"          Innings Complete\n";
        cout<<"*****\n";
        show();
        getch();
        return;
    }

    else
    {
        getch();
        show();
    }
}

```

```

/***** Wicket Fallen *****/

```

```

void myscore::wicket(void)
{
    int o_type,new_player;
    cout<<"\n\n*****\n";
    cout<<"          Wicket\n";
    cout<<"*****\n";
    cout<<"\nOut type? (Bold-1:Caught-2:Run_out:3).....";
    cin>>o_type;
    Out++;
    Ball_count++;
    team1.Set_Ball(player1);

    if(Out>=Max_out)
    {
        cout<<"\n\n*****\n";
        cout<<"          Innings Complete\n";
        cout<<"*****\n";
        show();
        team1.Set_Status(player1,o_type+1);
        getch();
        return;
    }

    cout<<"\nNew Batsmans no: ";
    cin>>new_player;
    team1.Set_Out(player1,o_type+1,new_player);
    player1=new_player;

    if(Ball_count==6)
    {
        over_complete();
        return;
    }

    getch();
    show();
}

int main(void)
{

```

```
myscore MS;  
MS.show();  
}
```