{

1. Student:

```
#include<stdio.h>
#include<string.h>
struct student
{
        int roll_no;
        char name[20];
        student()
        {
                printf("\n\ndefault constructor called\n");
                this->roll_no=0;
                strcpy(this->name,"not_given");
        }
        student(int r,char* n)
        {
                printf("\n\nparameterised constructor called\n");
                this->roll_no=r;
                strcpy(this->name,n);
        }
        void setroll(int r)                //setters(mumtator)
        {
                this->roll_no=r;
        }
        void setname(const char* n)
        {
                strcpy(this->name,n);       //setters(mumtator)
        }
        int getroll()               //getters(accessors)
        {
                return this->roll_no;
        }
        char* getname()             //getters(accessors)
        {
                return this->name;
        }
        void display()
        {
                printf("\nroll no %d is %s\n",this->roll_no,this->name);
        }
};
int main()
{
        student s1;
        int roll_no;
        char name[20];
        s1.display();
        printf("\nenter roll no of student: ");
        scanf("%d",&roll_no);
        printf("\nenter name of the student: ");
        scanf("%s",&name);
        s1.setroll(roll_no);        //member function
        s1.setname(name);       //member function
        printf("\nafter setting value\n");
        s1.display();       //member function
        printf("\ngetter\n");
        printf("\nroll no: %d \nname: %s",s1.getroll(),s1.getname());
        student s3;
        s3.setroll(10);
        s3.setname("sachin");
        printf("\nafter setting value\n");
        s3.display();       //member function
        printf("\ngetter\n");
        printf("\nroll no: %d \nname: %s",s3.getroll(),s3.getname());
        student s2(42,"pragati");           //member function
        s2.display();       //member function
        printf("\ngetter\n");
        printf("\nroll no: %d \nname: %s",s2.getroll(),s2.getname());
        return 0;
}
```

2. Employee:

```
#include<stdio.h>
#include<string.h>
struct employee
{
        int emp_id;
```

```
char name[20];
double salary;
employee()
{
	printf("\n\ndefault constructor called\n");
	this->emp_id=0;
	strcpy(this->name,"not_given");
	this->salary=0;
}
employee(int i,const char* n,double s)
{
	printf("\n\nparameterised called\n");
	this->emp_id=i;
	strcpy(this->name,n);
	this->salary=s;
}
void setid(int i)  //setters(mutators)
{
	this->emp_id=i;
}
void setname(const char* n)	//setters(mutators)
{
	strcpy(this->name,n);
}
void setsalary(double s) //setters(mutators)
```

```
        {
                this->salary=s;
        }
        int getid()          //getters(accessors)
        {
                return this->emp_id;
        }
        char* getname()            //getters(accessors)
        {
                return this->name;
        }
        double getsalary()          //getters(accessors)
        {
                return this->salary;
        }
        void display()
        {
                printf("\nemployees detail: \nid: %d \tname: %s  \tsalary: %.2lf\n",this->emp_id,this->name,this->salary);
        }
};
int main()
{
        employee e1;
        int emp_id;
        char name[20];
        double salary;

        e1.display();     //member function

        printf("\nenter employee id:\n");

        scanf("%d",&emp_id);

        printf("enter employee name: \n");

        scanf("%s",&name);

        printf("enter employee salary: \n");

        scanf("%lf",&salary);

        e1.setid(emp_id);         //member function

        e1.setname(name);       //member function

        e1.setsalary(salary);       //member function

        printf("\nafter setting values\n");

        e1.display();     //member function

        printf("\nemployees detail: \nid: %d \nname: %s  \nsalary:
%.2lf",e1.getid(),e1.getname(),e1.getsalary());

        employee e3;

        e3.display();

        e3.setid(401);    //member function

        e3.setname("sachin");   //member function

        e3.setsalary(60000);      //member function

        printf("\nafter setting values\n");

        e3.display();     //member function

        printf("\nemployees detail: \nid: %d \nname: %s  \nsalary:
%.2lf",e3.getid(),e3.getname(),e3.getsalary());

        employee e2(22,"pragati",50000);          //member function

        e2.display();     //member function
```

```
    printf("\ngetter\n");
    printf("\nemployees detail: \nid: %d \nname: %s  \nsalary: %.2lf",e2.getid(),e2.getname(),e2.getsalary());
    return 0;
}
```

3. Sales manager:

```
#include<stdio.h>
#include<string.h>
struct sales_man
{
    int id,target;
    char name[20];
    double salary,intensive;
    sales_man()
    {
        printf("\n\ndefault constructor called\n");
        this->id=0;
        strcpy(this->name,"not_given");
        this->salary=0;
        this->target=0;
        this->intensive=0;
    }
    sales_man(int i,const char* n,double s,int t,int in)
    {
        printf("\n\nparameterised constructor called\n");
        this->id=i;
}

        strcpy(this->name,n);
        this->salary=s;
        this->target=t;
        this->intensive=in;
}
void setid(int i)  //setters(mutator)
{
        this->id=i;
}
void setname(const char* n)             //setters(mutator)
{
        strcpy(this->name,n);
}
void setsalary(double s)        //setters(mutator)
{
        this->salary=s;
}
void settarget(int t)           //setters(mutator)
{
        this->target=t;
}
void setintense(double in)              //setters(mutator)
{
        this->intensive=in;
}
```

```
    int getid()                 //getters(accessor)
    {
        return this->id;
    }
    char* getname()                 //getters(accessor)
    {
        return this->name;
    }
    double getsalary()              //getters(accessor)
    {
        return this->salary;
    }
    int gettarget()             //getters(accessor)
    {
        return this->target;
    }
    double getintense()             //getters(accessor)
    {
        return this->intensive;
    }
    void display()
    {
        printf("\nsales managers details:\nid: %d\tname: %s\tsalary: %.2lf\ttarget:
%d\tintensive: %.2lf\n",this->id,this->name,this->salary,this->target,this->intensive);
    }
};
```

```
int main()
{
        sales_man m1;
        int id,target;
        char name[20];
        double salary,intensive;
        m1.display();    //member function
        printf("enter sale managers id:\n");
        scanf("%d",&id);
        printf("\nenter the name of sales manager:\n");
        scanf("%s",name);
        printf("\nenter salary of sales manager:\n");
        scanf("%lf",&salary);
        printf("\nenter target of sales manager:\n");
        scanf("%d",&target);
        printf("\nenter intensive for target completion:\n");
        scanf("%lf",&intensive);
        m1.setid(id);    //member function
        m1.setname(name);               //member function
        m1.setsalary(salary);           //member function
        m1.settarget(target);
        m1.setintense(intensive);       //member function
        printf("\nafter setting values\n");
        m1.display();    //member function
        printf("\ngetter\nsales managers details:\nid: %d\tname: %s\nsalary: %.2lf\ttarget: %d\tintensive: %.2lf\n",m1.getid(),m1.getname(),m1.getsalary(),m1.gettarget(),m1.getintense());
        sales_man m3;

        m3.display();

        m3.setid(101);   //member function

        m3.setname("sachin");               //member function

        m3.setsalary(60000);                //member function

        m3.settarget(40);

        m3.setintense(4500);     //member function

        printf("\nafter setting values\n");

        m3.display();     //member function

        printf("\ngetter\nsales managers details:\nid: %d\tname: %s\nsalary: %.2lf\ttarget: %d\tintensive: %.2lf\n",m1.getid(),m1.getname(),m1.getsalary(),m1.gettarget(),m1.getintense());

        sales_man m2(22,"pragati",50000,45,4500);

        m2.display();     //member function

        printf("\ngetter\nsales managers details:\nid: %d\tname: %s\nsalary: %.2lf\ttarget: %d\tintensive: %.2lf\n",m2.getid(),m2.getname(),m2.getsalary(),m2.gettarget(),m2.getintense());

        return 0;

}
```

4. Admin:

```
#include<stdio.h>

#include<string.h>

struct admin

{

        int id;

        char name[20];

        double salary,allowance;

        admin()
```

```
{
        printf("\n\ndefault constructor called\n");
        this->id=0;
        strcpy(this->name,"not_given");
        this->salary=0;
        this->allowance=0;
}
admin(int i,const char* n,double s,double a)
{
        printf("\n\nparameterised constructor called\n");
        this->id=i;
        strcpy(this->name,n);
        this->salary=s;
        this->allowance=a;
}
void setid(int i)  //setters(mutator)
{
        this->id=i;
}
void setname(const char* n)                    //setters(mutator)
{
        strcpy(this->name,n);
}
void setsalary(double s) //setters(mutator)
{
        this->salary=s;
}
void setallow(double a)         //setters(mutator)
{
        this->allowance=a;
}
int getid()             //getters(accessor)
{
        return this->id;
}
char* getname()             //getters(accessor)
{
        return this->name;
}
double getsalary()          //getters(accessor)
{
        return this->salary;
}
double getallow()       //getters(accessor)
{
        return this->allowance;
}
void display()
{
        printf("\nadmins details:\nid: %d\tname: %s\tsalary: %.2lf\tallowance: %.2lf\n",this->id,this->name,this->salary,this->allowance);
```

```
	}
};
int main()
{
	admin a1;
	int id;
	char name[20];
	double salary,allowance;
	a1.display();	//member function
	printf("enter admin id:\n");
	scanf("%d",&id);
	printf("\nenter name of the admin:\n");
	scanf("%s",name);
	printf("\nenter salary of admin:\n");
	scanf("%lf",&salary);
	printf("\nallowance for admin:\n");
	scanf("%lf",&allowance);
	a1.setid(id);	//member function
	a1.setname(name);	//member function
	a1.setsalary(salary);	//member function
	a1.setallow(allowance); //member function
	printf("\nafter setting values\n");
	a1.display();	//member function
	printf("\ngetter\n");
	printf("\nadmins details:\nid: %d\nname: %s\nsalary: %.2lf\nallowance: %.2lf\n",a1.getid(),a1.getname(),a1.getsalary(),a1.getallow());
```

```
    admin a3;
    a3.display();
    a3.setid(101);    //member function
    a3.setname("sachin");   //member function
    a3.setsalary(60000);      //member function
    a3.setallow(6000);         //member function
    printf("\nafter setting values\n");
    a3.display();      //member function
    printf("\ngetter\n");
    printf("\nadmins details:\nid: %d\nname: %s\nsalary: %.2lf\nallowance: %.2lf\n",a3.getid(),a3.getname(),a3.getsalary(),a3.getallow());
    admin a2(101,"pragati",50000,4500);
    a2.display();      //member function
    printf("\ngetter\n");
    printf("\nadmins details:\nid: %d\nname: %s\nsalary: %.2lf\nallowance: %.2lf\n",a2.getid(),a2.getname(),a2.getsalary(),a2.getallow());
    return 0;
}
```

5. HR manager:

```
#include<stdio.h>
#include<string.h>
struct hr_manager
{
    int id;
    char name[20];
    double salary,commission;
```

```
hr_manager()

{

            printf("\n\ndefault constructor called\n");

            this->id=0;

            strcpy(this->name,"not_given");

            this->salary=0;

            this->commission=0;

}

hr_manager(int i,const char* n,double s,double c)

{

            printf("\n\nparameterised constructor called\n");

            this->id=i;

            strcpy(this->name,n);

            this->salary=s;

            this->commission=c;

}

void setid(int i)  //setters(mutator)

{

            this->id=i;

}

void setname(const char* n)                    //setters(mutator)

{

            strcpy(this->name,n);

}

void setsalary(double s) //setters(mutator)
```

```
{
        this->salary=s;
}
void setcomm(double c)          //setters(mutator)
{
        this->commission=c;
}
int getid()                     //getters(accessor)
{
        return this->id;
}
char* getname()                     //getters(accessor)
{
        return this->name;
}
double getsalary()                  //getters(accessor)
{
        return this->salary;
}
double getcomm()        //getters(accessor)
{
        return this->commission;
}
void display()
{
		printf("\nHR Managers detail: \nid: %d\tName: %s\tSalary: %.2lf\tCommission: %.2lf\n",this->id,this->name,this->salary,this->commission);
	}
};
int main()
{
	hr_manager h1;
	int id;
	char name[20];
	double salary,commission;
	h1.display();	//member function
	printf("\nenter hr managers id:\n");
	scanf("%d",&id);
	printf("\nenter name of hr manager:\n");
	scanf("%s",name);
	printf("\nenter salary of hr manager:\n");
	scanf("%lf",&salary);
	printf("\nenter commission for hr manager:\n");
	scanf("%lf",&commission);
	h1.setid(id);	//member function
	h1.setname(name);	//member function
	h1.setsalary(salary);	//member function
	h1.setcomm(commission);	//member function
	printf("\nafter setting values\n");
	h1.display();	//member function
```

```
    printf("\ngetter\nHR Managers detail: \nid: %d\nName: %s\nSalary: %.2lf\nCommission: %.2lf\n",h1.getid(),h1.getname(),h1.getsalary(),h1.getcomm());
    hr_manager h3;
    h3.display();
    h3.setid(101);    //member function
    h3.setname("sachin");   //member function
    h3.setsalary(60000);      //member function
    h3.setcomm(6000);       //member function
    printf("\nafter setting values\n");
    h3.display();      //member function
    printf("\ngetter\nHR Managers detail: \nid: %d\nName: %s\nSalary: %.2lf\nCommission: %.2lf\n",h3.getid(),h3.getname(),h3.getsalary(),h3.getcomm());
    hr_manager h2(202,"pragati",50000,5000);        //member function
    h2.display();      //member function
    printf("\ngetter\nHR Managers detail: \nid: %d\nName: %s\nSalary: %.2lf\nCommission: %.2lf\n",h2.getid(),h2.getname(),h2.getsalary(),h2.getcomm());
    return 0;
}
```

6. Date:

```
#include<stdio.h>
struct date
{
    int day,month,year;
    date()
    {
        printf("\n\ndefault constructor called\n");
        this->day=0;
        this->month=0;

        this->year=0;

}

date(int d,int m,int y)

{

        printf("\n\nparameterised constructor called\n");

        this->day=d;

        this->month=m;

        this->year=y;

}

void setday(int d)      //setter(mutator)

{

        this->day=d;

}

void setmonth(int m)    //setter(mutator)

{

        this->month=m;

}

void setyear(int y)             //setter(mutator)

{

        this->year=y;

}

int getday()    //getters(accessor)

{

        return this->day;
        }
        int getmonth()                //getters(accessor)
        {
                return this->month;
        }
        int getyear()                 //getters(accessor)
        {
                return this->year;
        }
        void display()
        {
                printf("\n\ndate is: \n %d/%d/%d\n",this->day,this->month,this->year);
        }
};
int main()
{
        date d1;
        int day,month,year;
        d1.display();      //member function
        printf("\nenter date: ");
        scanf("%d",&day);
        printf("\nenter month: ");
        scanf("%d",&month);
        printf("\nenter year: ");
        scanf("%d",&year);
```

```
	d1.setday(day);		//member function
	d1.setmonth(month);		//member function
	d1.setyear(year);		//member function
	printf("\nafter setting values\n");
	d1.display();	//member function
	printf("\ngetter\ndate: %d\nmonth: %d\nyear: %d\n",d1.getday(),d1.getmonth(),d1.getyear());
	date d3;
	d3.display();
	d3.setday(4);		//member function
	d3.setmonth(10);		//member function
	d3.setyear(2018);		//member function
	printf("\nafter setting values\n");
	d1.display();	//member function
	printf("\ngetter\ndate: %d\nmonth: %d\nyear: %d\n",d1.getday(),d1.getmonth(),d1.getyear());
	date d2(23,4,2001);		//member function
	d2.display();
	printf("\ngetter\ndate: %d\nmonth: %d\nyear: %d\n",d2.getday(),d2.getmonth(),d2.getyear());
	return 0;
}
```

7. Time:

```
#include<stdio.h>
struct time
{
	int hr,min,sec;
	time()
}
{
        printf("\n\ndefault constructor called\n");
        this->hr=-1;
        this->min=-1;
        this->sec=-1;
}
time(int h,int m,int s)
{
        printf("\n\nparameterised constructor called\n");
        this->hr=h;
        this->min=m;
        this->sec=s;
}
void sethour(int h)                 //setter(mutator)
{
        this->hr=h;
}
void setmin(int m)                  //setter(mutator)
{
        this->min=m;
}
void setsec(int s)                  //setter(mutator)
{
        this->sec=s;
}
```

```
        int gethr()                  //getter(accessor)

        {

                return this->hr;

        }

        int getmin()      //getter(accessor)

        {

                return this->min;

        }

        int getsec()      //getter(accessor)

        {

                return this->sec;

        }

        void display()

        {

                printf("\ntime is: %d:%d:%d\n",this->hr,this->min,this->sec);

        }

};

void display(time*);

int main()

{

        time t1;

        int hr,min,sec;

        int r,q;

        t1.display();     //member function

        printf("\nenter hours:\n");
```

```
scanf("%d",&hr);
printf("\nenter minuits:\n");
scanf("%d",&min);
printf("\nenter seconds:\n");
scanf("%d",&sec);
if(sec>=60)
{
        r=sec%60;
        q=sec/60;
        sec=r;
        min=min+q;
}
if(min>=60)
{
        r=min%60;
        q=min/60;
        min=r;
        hr=hr+q;
}
t1.setsec(sec);          //member function
t1.setmin(min);          //member function
t1.sethour(hr);          //member function
printf("\nafter setting value\n");
t1.display();    //member function
printf("\ngetter\nhour: %d\nmin: %d\nsec: %d\n",t1.gethr(),t1.getmin(),t1.getsec());
```

```
    time t3;
    t3.display();
    t3.setsec(40);          //member function
    t3.setmin(56);          //member function
    t3.sethour(7);          //member function
    printf("\nafter setting value\n");
    t3.display();   //member function
    printf("\ngetter\nhour: %d\nmin: %d\nsec: %d\n",t3.gethr(),t3.getmin(),t3.getsec());
    time t2(10,49,55);              //member function
    t2.display();   //member function
    printf("\ngetter\nhour: %d\nmin: %d\nsec: %d\n",t1.gethr(),t1.getmin(),t1.getsec());
    return 0;
}
```

8. Distance:

```
#include<stdio.h>
struct distance
{
    int feet,inch;
    distance()
    {
        printf("\n\ndefault constructor called\n");
        this->feet=-1;
        this->inch=-1;
    }
    distance(int f,int i)
}

{
        printf("\n\nparameterised constructor called\n");
        this->feet=f;
        this->inch=i;
}
void setfeet(int f)         //setter(mutator)
{
        this->feet=f;
}
void setinch(int i)                 //setter(mutator)
{
        this->inch=i;
}
int getfeet()     //getter(accessor)
{
        return this->feet;
}
int getinch()     //getter(accessor)
{
        return this->inch;
}
void display()
{
        printf("\ndistance is: %d feet and %d inches\n",this->feet,this->inch);
}
```

```
};

int main()

{

        distance d1;

        int feet,inch;

        d1.display();     //member function

        printf("\nenter distance in feet:\n");

        scanf("%d",&feet);

        printf("\nenter distance in inch:\n");

        scanf("%d",&inch);

        d1.setfeet(feet);          //member function

        d1.setinch(inch);          //member function

        printf("\nafter setting values\n");

        d1.display();     //member function

        printf("\ngetter\nfeet: %d\ninch: %d\n",d1.getfeet(),d1.getinch());

        distance d3;

        d3.display();

        d3.setfeet(6);    //member function

        d3.setinch(2);    //member function

        printf("\nafter setting values\n");

        d3.display();     //member function

        printf("\ngetter\nfeet: %d\ninch: %d\n",d3.getfeet(),d3.getinch());

        distance d2(5,2);          //member function

        d2.display();     //member function

        printf("\ngetter\nfeet: %d\ninch: %d\n",d2.getfeet(),d2.getinch());
```

```
    {

    return 0;
}
```

9. Complex:

```
#include<stdio.h>
struct complex
{
    int real,imag;
    complex()
    {
        printf("\n\ndefault constructor called\n");
        this->real=0;
        this->imag=0;
    }
    complex(int r,int i)
    {
        printf("\n\nparameterised constructor called\n");
        this->real=r;
        this->imag=i;
    }
    void setreal(int r)                 //setters(mutator)
    {
        this->real=r;
    }
    void setimg(int i)          //setters(mutator)
    {
```

```
		this->imag=i;
	}
	int getreal()			//getters(accessor)
	{
		return this->real;
	}
	int getimag()	//getters(accessor)
	{
		return this->imag;
	}
	void display()
	{
		printf("\ncomplex number: %d+%di\n",this->real,this->imag);
	}
};
int main()
{
	complex c1;
	c1.display();	//member function
	int real,imag;
	printf("\nenter real part of complex number:\n");
	scanf("%d",&real);
	printf("\nenter imaginary part of complex number:\n");
	scanf("%d",&imag);
	c1.setreal(real); //member function
```

```
        c1.setimg(imag);            //member function

        printf("\nafter setting values\n");

        c1.display();               //member function

        printf("\ngetter\n");

        printf("\ncomplex number: %d+%di\n",c1.getreal(),c1.getimag());

        complex c3;

        c3.display();

        c3.setreal(15);   //member function

        c3.setimg(3);     //member function

        printf("\nafter setting values\n");

        c3.display();               //member function

        printf("\ngetter\n");

        printf("\ncomplex number: %d+%di\n",c3.getreal(),c3.getimag());

        complex c2(10,2);           //member function

        c2.display();     //member function

        printf("\ngetter\n");

        printf("\ncomplex number: %d+%di\n",c2.getreal(),c2.getimag());

        return 0;

}
```

10. Department:

```
#include<stdio.h>

#include<string.h>

struct dept

{

        int did;
}

char dname[20];

dept()

{

	printf("\nDefault constructor called\n");

	this->did=0;

	strcpy(this->dname,"not_given");

}

dept(int i,char* n)

{

	printf("\nParameterised constructor called\n");

	this->did=i;

	strcpy(this->dname,n);

}

void setid(int i)

{

	this->did=i;

}

void setname(char* n)

{

	strcpy(this->dname,n);

}

int getid()

{

	return this->did;

}
```

```
		char* getname()

		{

			return this->dname;

		}

		void display()

		{

			printf("\ndepartment details:\nDept id: %d\tName: %s\n",this->did,this->dname);

		}

};

int main()

{

		dept d1;

		d1.display();

		int id;

		char name[20];

		printf("\nEnter department id:\n");

		scanf("%d",&id);

		printf("\nEnter department name:\n");

		scanf("%s",name);

		d1.setid(id);

		d1.setname(name);

		printf("\nafter setting value\n");

		d1.display();

		printf("\n*getters*\ndept id: %d\ndept name: %s\n",d1.getid(),d1.getname());

		dept d2;
```

```
    d2.display();

    d2.setid(10);

    d2.setname("comp_sci");

    printf("\nafter setting value\n");

    d2.display();

    printf("\n*getters*\ndept id: %d\ndept name: %s\n",d2.getid(),d2.getname());

    dept d3(20,"mathematics");

    d3.display();

    printf("\n*getters*\ndept id: %d\ndept name: %s\n",d3.getid(),d3.getname());

    return 0;

}
```

11. Player:

```
#include<stdio.h>

#include<string.h>

struct player

{

    int jersey;

    char pname[20],sname[20];

    player()

    {

        printf("\nDefault constructor called\n");

        this->jersey=0;

        strcpy(this->pname,"not_given");

        strcpy(this->sname,"not_given");

    }
```

```
{

player(int i,char* n,char* s)

{

        printf("\nParameterised constructor called\n");

        this->jersey=i;

        strcpy(this->pname,n);

        strcpy(this->sname,s);

}

void setjersey(int j)

{

        this->jersey=j;

}

void setpname(char* n)

{

        strcpy(this->pname,n);

}

void setsport(char* s)

{

        strcpy(this->sname,s);

}

int getjersey()

{

        return this->jersey;

}

char* getname()

{
```

```
                return this->pname;
        }
        char* getsport()
        {
                return this->sname;
        }
        void display()
        {
                printf("\nplayers detail: \njersey no: %d\tName: %s\tSport: %s\n",this->jersey,this->pname,this->sname);
        }
};
int main()
{
        player p1;
        p1.display();
        int j;
        char name[20],sport[20];
        printf("\nEnter jersey no:\n");
        scanf("%d",&j);
        printf("\nEnter player name:\n");
        scanf("%s",name);
        printf("\nEnter sport name:\n");
        scanf("%s",sport);
        p1.setjersey(j);
        p1.setpname(name);
        p1.setsport(sport);

        printf("\nafter setting value\n");

        p1.display();

        printf("\n*gatter*\njersey no: %d\nplayer name: %s\nsport name: %s\n",p1.getjersey(),p1.getname(),p1.getsport());

        player p2;

        p2.display();

        p2.setjersey(7);

        p2.setpname("M.S.Dhoni");

        p2.setsport("cricket");

        printf("\nafter setting value\n");

        p2.display();

        printf("\n*gatter*\njersey no: %d\nplayer name: %s\nsport name: %s\n",p2.getjersey(),p2.getname(),p2.getsport());

        player p3(7,"Ronaldo","football");

        p3.display();

        printf("\n*gatter*\njersey no: %d\nplayer name: %s\nsport name: %s\n",p3.getjersey(),p3.getname(),p3.getsport());

        return 0;

}
```

12. Team:

```
#include<stdio.h>

#include<string.h>

struct team

{

        int tid,no_emp;

        char tname[20],thead[20];
```

```
team()
{
	printf("\nDefault constructor called\n");
	this->tid=0;
	strcpy(this->tname,"not_given");
	strcpy(this->thead,"not_given");
	this->no_emp=0;
}
team(int i,char* n,char* h,int ne)
{
	printf("\nParameterised constructor called\n");
	this->tid=i;
	strcpy(this->tname,n);
	strcpy(this->thead,h);
	this->no_emp=ne;
}
void setid(int i)
{
	this->tid=i;
}
void setname(char* n)
{
	strcpy(this->tname,n);
}
void sethead(char* h)
```

```
{
	strcpy(this->thead,h);
}
void setemp(int ne)
{
	this->no_emp=ne;
}
int getid()
{
	return this->tid;
}
char* getname()
{
	return this->tname;
}
char* gethead()
{
	return this->thead;
}
int getemp()
{
	return this->no_emp;
}
void display()
{
		printf("\nTeam details:\nteam id: %d\tName: %s\nHead of team: %s\t No. of employee: %d\n",this->tid,this->tname,this->thead,this->no_emp);
	}
};
int main()
{
	team t1;
	int id,emp;
	char name[20],head[20];
	t1.display();
	printf("\nEnter team id:\n");
	scanf("%d",&id);
	printf("\nEnter team name:\n");
	scanf("%s",name);
	printf("\nEnter name of team head:\n");
	scanf("%s",head);
	printf("\nEnter no of employees working in that team:\n");
	scanf("%d",&emp);
	t1.setid(id);
	t1.setname(name);
	t1.sethead(head);
	t1.setemp(emp);
	printf("\nafter setting value:\n");
	t1.display();
	printf("\n*getter*\nteam id: %d\nteam name: %s\nteam head: %s\nno. of employees: %d\n",t1.getid(),t1.getname(),t1.gethead(),t1.getemp());
	team t2;
	t2.display();
	t2.setid(11);
	t2.setname("Quality_assurance");
	t2.sethead("Prakruti");
	t2.setemp(15);
	printf("\nafter setting value:\n");
	t2.display();
	printf("\n*getter*\nteam id: %d\nteam name: %s\nteam head: %s\nno. of employees: %d\n",t2.getid(),t2.getname(),t2.gethead(),t2.getemp());
	team t3(22,"front_end","Pragati",25);
	t3.display();
	printf("\n*getter*\nteam id: %d\nteam name: %s\nteam head: %s\nno. of employees: %d\n",t3.getid(),t3.getname(),t3.gethead(),t3.getemp());
}
```