Assignment no 1

1. Student (rollNo, name, marks)

```
#include<stdio.h>
#include<string.h>
struct Student{
	int roll;
	char name[20];
	Student(){
		printf("default constructor called\n");
		this->roll=0;
		strcpy(this->name,"not given");
	}
	Student(int r,char* nm){
		printf("paramterised constructor called\n");
		this->roll=r;
		strcpy(this->name,nm);

	}

	void setroll(int a){
		this->roll=a;
	}
	void setname(char* nm){
		strcpy(this->name,nm)
```

```
}
        int getroll(){
                return this->roll;
        }
        char* getname(){
                return this->name;
        }
        void display(){
                printf("id is %d",this->roll);
                printf(" name is %s\n",this->name);
        }
};
int main(){
        Student s1;
        s1.setroll(10);
        s1.setname("sachin");
        //printf("%d",s1.getroll());
//      printf(" %s\n",s1.getname());
   s1.display();
        Student s2(3,"raina");
        s2.setroll(18);
        s2.setname("virat");
        //printf("%d",s2.getroll());
        //printf(" %s",s2.getname());
        s2.display();
}
```

2 . Employee (id, name, salary)

```
#include <stdio.h>
#include<string.h>
struct emp{
        int emp_id;
        char name[20];
        double salary;

        emp(){
                printf("default constructor called\n");
                this->emp_id=0;
                strcpy(this->name,"not given");
                this->salary=0;
        }
        emp(int a,char*nm,double s){
                printf("parameterised constructor called\n");
                this->emp_id=a;
                strcpy(this->name,nm);
                this->salary=s;

        }


 void setid(int a){
      this->emp_id=a;
```

```
}
void setname(char* nm){
        strcpy(this->name,nm);
}
void setsalary(double a){
        this->salary=a;
}
void display(){
        printf("id= %d",this->emp_id);
        printf(" name= %s",this->name);
        printf(" salary= %.2lf",this->salary);
}
int getid(){
        return this->emp_id;
}
char* getname(){
        return this->name;
}
double getsalary(){
        return this->salary;
}

};

int main() {

            emp e;
      e.setid(1);
```

```
        e.setname("sandesh");
        e.setsalary(1290.23);
        e.display();
        printf("\n");
        //printf("%d",e.getid());
        //printf(" %s",e.getname());
        //printf(" %.2lf",e.getsalary());
        emp e1(18,"virat",5000);
        e1.setid(2);
        e1.setname("roman");
        e1.setsalary(1200.10);
        e1.display();
        //printf("id=%d",e1.getid());
       // printf("name=%s",e1.getname());
       // printf("salary=%.2lf",e1.getsalary());


}
```

3 .Admin (id, name, salary, allowance)

```
 #include<stdio.h>
#include<string.h>
struct admin{
        int id;
        char name[20];
```

```
double salary;
double allowance;

admin(){
        printf("default constructor called\n");
        this->id=0;
        strcpy(this->name,"not given");
        this->salary=0.00;
        this->allowance=0.00;
}
admin(int a,char*nm){
        printf("parameterised constructor called\n");
        this->id=0;
        strcpy(this->name,"not given");
}
void setid(int a){
        this->id=a;
}
void setname(char* nm){
        strcpy(this->name,nm);
}
void setsalary(double a){
        this->salary=a;
}
void setallowance(double b)
{
        this->allowance=b;
}
int getid(){
```

```
		return this->id;
	}
	char* getname(){
		return this->name;
	}
	double getsalary(){
		return this->salary;
	}
	double getallowance(){
		return this->allowance;
	}
	void display(){
		printf("id= %d name = %s salary = %.2lf allowance =%.2lf\n",this->id,this->name,this->salary,this->allowance);
	}

};
int main(){
	admin a;
	a.setid(1);
	a.setname("sandesh");
	a.setsalary(11200.20);
	a.setallowance(1200.20);
	printf("id= %d name= %s salary= %.2lf allowance= %lf\n",a.getid(),a.getname(),a.getsalary(),a.getcommission());
//	a.display();
	admin a1(2,"ram");
	a1.setid(2);
	a1.setname("ram");
```

```
	a1.setsalary(111200.20);
	a1.setallowance(12400.20);
	printf("id= %d name= %s salary= %.2lf allowance=
%lf\n",a1.getid(),a1.getname(),a1.getsalary(),a1.getallowance());
	//a1.display();

}
```

4. HR (id, name, salary, commission)

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

	hr(){
		printf("default constructor called\n");
		this->id=0;
		strcpy(this->name,"not given");
		this->salary=0.00;
		this->commission=0.00;
	}
	hr(int a,char*nm){
```

```
        printf("parameterised constructor called\n");
        this->id=0;
        strcpy(this->name,"not given");
}
void setid(int a){
        this->id=a;
}
void setname(char* nm){
        strcpy(this->name,nm);
}
void setsalary(double a){
        this->salary=a;
}
void setcommission(double b)
{
        this->commission=b;
}
int getid(){
        return this->id;
}
char* getname(){
        return this->name;
}
double getsalary(){
        return this->salary;
}
double getcommission(){
        return this->commission;
}
```

```
        void display(){
        printf("id= %d name = %s salary = %.2lf commission =%.2lf\n",this->id,this->name,this->salary,this->commission);
        }


};
int main(){
        hr a;
        a.setid(1);
        a.setname("sandesh");
        a.setsalary(11200.20);
        a.setcommission(1200.20);
        printf("id= %d name= %s salary= %.2lf commission= %lf\n",a.getid(),a.getname(),a.getsalary(),a.getcommission());
//      a.display();
        hr a1(2,"ram");
        a1.setid(2);
        a1.setname("ram");
        a1.setsalary(111200.20);
        a1.setcommission(12400.20);
        printf("id= %d name= %s salary= %.2lf commission= %lf\n",a1.getid(),a1.getname(),a1.getsalary(),a1.getcommission());
        //a1.display();


}
```

5. SalesManager (id, name, salary, incentive, target)

```
#include<stdio.h>
#include<string.h>
struct sales_manager{
	int id;
	char name[20];
	double salary;
	int target;
	double incentive;

	sales_manager(){
		printf("default constructor called\n");
		this->id=0;
		strcpy(this->name,"not given");
		this->salary=00.00;
		this->target=0;
		this->incentive=0;
	}
	sales_manager(int a,char* nm){
		printf("parameterised constructor called\n");
		this->id=a;
		strcpy(this->name,nm);
	}
	void setid(int a){
		this->id=a;
```

```
}
void setname(char* nm){
        strcpy(this->name,nm);
}
void setsalary(double a){
        this->salary=a;
}
void settarget(int a){
        this->target=a;
}
void setincentive(double a){
        this->incentive=a;
}
int getid(){
        return this->id;
}
char* getname(){
        return this->name;
}
double getsalary(){
        return this->salary;
}
int gettarget(){
        return this->target;
}
double getincentive(){
        return this->incentive;
}
void display(){
```

```
                printf("id = %d name= %s salary= %.2lf target= %.2lf incentive= %.2lf",this->id,this->name,this->salary,this->target,this->incentive);
        }



};
int main(){
        sales_manager s1;
        s1.setid(1);
        s1.setname("sandesh");
        s1.setsalary(1200);
        s1.settarget(12);
        s1.setincentive(435);
        s1.display();
        sales_manager s2(0,"not given");
        s2.setid(2);
        s2.setname("ram");
        s2.setsalary(1000);
        s2.settarget(50);
        s2.setincentive(435);
        s2.display();




}
```

6. Date (date, month, year)

```
#include<stdio.h>
struct date{
	int day;
	int month;
	int year;

	void set_day(int d){
		this->day=d;
	}
	void set_month(int m){
		this->month=m;
	}
	void set_year(int y){
		this->year=y;
	}
	void display(){
		printf("day= %d month= %d year= %d",this->day,this->month,this->year);
	}
	int getday(){
		return this->day;
	}
	int getmonth(){
		return this->month;
	}
	int getyear(){
		return this->year;
```

```
        }

};
int main(){
        date d1,d2;
        d1.set_day(7);
        d1.set_month(3);
        d1.set_year(2001);
        d1.display();
        printf("\n");
        d2.set_day(10);
        d2.set_month(8);
        d2.set_year(2001);
        d2.display();
}
```

7. Time (hour, min, sec)

```
#include <stdio.h>
struct time{
        int hour;
        int min;
        int sec;
        time(){
                printf("default constructor called\n");
                 this->hour=0;
                 this->min=0;
```

```
        this->sec=0;
    }
    time(int h,int m,int s){
        printf("parameterised constructor called\n");
        this->hour=0;
        this->min=0;
        this->sec=0;
    }
void sethour(int a){
    this->hour=a;
}
void setmin(int a){
    this->min=a;
}
void setsec(int a){
    this->sec=a;
}
int gethour(){
    return this->hour;
}
int getmin(){
    return this->min;
}
int getsec(){
    return this->sec;
}
void display(){

     printf("hour=%d min=%d sec=%d\n",this->hour,this->min,this->sec);
```

```
}



};

int main() {
        time d;
        d.sethour(5);
        d.setmin(39);
        d.setsec(23);
        d.display();
        time d2(2,45,30);
        d2.display();

}
```

8 .Distance (feet, inch)

```
        #include<stdio.h>
struct distance{
        int feet;
        int inch;

        distance(){
                printf("default constructor called\n");
                this->feet=0;
```

```
        this->inch=0;
    }
    distance(int a,int b){
        printf("parameterised called\n");
        this->feet=a;
        this->inch=b;
    }
    void setfeet(int a){
        this->feet=a;
    }
    void setinch(int b){
        this->inch=b;
    }
    int getfeet(){
        return this->feet;
    }
    int getinch(){
        return this->inch;
    }
    void display(){
        printf("feet= %d inch= %d\n",this->feet,this->inch);
    }

};
int main(){
    distance d1;
    d1.setfeet(5);
    d1.setinch(2);
```

```
        d1.display();
        distance d2(0,0);
        d2.setfeet(10);
        d2.setinch(20);
        d2.display();

}
```

9. Complex (real, imaginary)

```
#include<stdio.h>
struct complex{
        int real;
        int imaginary;
        complex(){
                printf("default constructor called\n");
                this->real=0;
                this->imaginary=0;
        }
        complex(int a,int i){
                printf("parameterised constructor called\n");
                this->real=a;
                this->imaginary=i;
        }
        void setreal(int a){
                this->real=1;
        }
```

```
        void setimaginary(int i){
                this->imaginary=i;
        }
        int getreal(){
                return this->real;
        }
        int getimaginary(){
                return this->imaginary;
        }
        void display(){
                printf("complex number %d+%di\n",real,imaginary);
        }

};
int main(){
        complex c1;
        c1.setreal(1);
        c1.setimaginary(5);
        c1.display();
        complex c2(0,0);
        c2.setreal(2);
        c2.setimaginary(8);
        c2.display();
}
```

10. Product (id, name, quantity, price)

```
#include<stdio.h
#include<string.h>
struct product{
        int id;
        char name[20];
        int quantity;
        double price;
        product(){
                printf("default constructor called\n");
                this->id=0;
                strcpy(this->name,"not given");
                this->quantity=0;
                this->price=0.00;
        }
        product(int a,char* nm){
                printf("paramaterised called\n");
                this->id=a;
                strcpy(this->name,nm);
        }
        void setid(int i){
    this->id=i;
        }
        void setname(char *nm){
                strcpy(this->name,nm);
        }
        void setquantity(int q){
                this->quantity=q;
```

```
}
        void setprice(double p){
                this->price=p;
        }
        int getid(){
                return this->id;
        }
        char* getname(){
                return this->name;
        }
        int getquantity(){
                return this->quantity;
        }
        void display(){
                printf("id= %d name= %s quantity= %d price= %.2lf\n",this->id,this->name,this->quantity,this->price);
        }

};
int main(){
        product p1;
        p1.setid(1);
        p1.setname("hp");
        p1.setquantity(5);
        p1.setprice(35000);
        p1.display();
        product p2(0,"not given");
        p2.setid(2);
        p2.setname("dell");
```

```
        p2.setquantity(5);
        p2.setprice(75000);
        p2.display();
}
```