Q 1:

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

struct Admin{    // here i create structure
        int id;
        char name[20];
        float salary;
        double allowance;

        Admin(){   // default contructor and value assignment
        printf("\nIm in Default Constructor !");
                id=0;
                strcpy(name,"Not Assigned");
                salary=0;
                allowance=0;
        }
        Admin(int id,char *name,float salary,double allowance){ // Parameterized Constructor
        printf("\n\nIm in Parameterised Constructor !");
                this->id=id;
                strcpy(this->name,name);
                this->salary=salary;
                this->allowance;
        }

        void setId(int id){      //here starting of Setter
                this->id=id;
        }
        void setName(char *name){
                strcpy(this->name,name);
        }
        void setSalary(float salary){
```

```
		this->salary=salary;
	}
	void setAllowance(double allowance){
		this->allowance=allowance;
	}

	void display()  //here display funtion
	{
		printf("\nId :%d",this->id);
		printf("\nName :%s",this->name);
		printf("\nSalary :%lf",this->salary);
		printf("\nAllowance :%f",this->allowance);
	}

};
int main(){
	Admin a1;   // object create and it will treat as default .
	a1.display();

	Admin a2;  // object created and we send parameter to contructor .
	a2.setId(1);
	a2.setName("Rahul");
	a2.setSalary(100);
	a2.setAllowance(1200);
	a2.display();

	Admin a3;   // object create and it will treat as default .
	a3.display();

	return 0;
}
```

Q 2:

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

typedef struct complex{
	int real,img;

	complex(){
		real=10;
		img=20;
		printf("\nDefault Constructor Called ");
	}

	complex(int real,int img)
	{
		this->real=real;
		this->img=img;
		printf("\nParametrised Constructor Called ");
	}

	void setReal(int real){
		this->real=real;
	}
	void setImg(int img){
		this->img=img;
	}

	void display(){
		printf("\n%d + %di",this->real,this->img);
	}
}complex;
```

```
int main(){
        complex c1;
        c1.display();

        complex c2(20,30);
        c2.display();

   complex c3;
   c3.setReal(11);
        c3.setImg(14);

   c3.display();

}
```

Q 3:

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

struct Employee{
        int id;
        char name[20];
        double salary ;

        Employee(){
                int id=0;
                strcpy(name,"not assigned ");
                salary=0;
                printf("\nThis is Default constructor !\n");
        }

        Employee(int id,char * name,double salary)
```

```
        {
                this->id=id;
                strcpy(this->name,name);
                this->salary=salary;
                printf("\nThis is Paramterised constructor !");
        }
        void setId(int id){
                this->id=id;
        }
        void setName(char *name){
                strcpy(this->name,name);
        }
        void setSalary(double salary){
                this->salary=salary;
        }
        void display()
        {
                printf("\nID is :%d\n",this->id);
                printf("Name is :%s\n",this->name);
                printf("Salary is :%lf\n",this->salary);
        }
};
int main(){
        Employee e1;
        e1.display();

        Employee e2(101,"Rahul",35000);
        e2.display();

        Employee e3;
        e3.setId(103);
        e3.setName("Devashree");
        e3.setSalary(12000);
```

```
        e3.display();
}
```

Q 4:

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

        HR(){           // here default constructor
                printf("\nIm in Default Constructor !\n");
                id=0;
                strcpy(name,"Rahul");
                salary=50;
                commission=50;
        }
        HR(int id,char *name ,float salary,double commission){    //here paramertrized constructor
           printf("\nIm in Parametrised Constructor !\n");
                this->id=id;
                strcpy(this->name,name);
                this->salary=salary;
                this->commission=commission;
        }

        void setId(int id){    // here starting of setter
                this->id=id;
        }
        void setName(char *name ){
                strcpy(this->name,name);
        }
```

```
	void setSalary(float salary){
		this->salary=salary;
	}
	void setCommission(double commision){
		this->commission=commission;
	}

	void display(){    // display function
		printf("\nId :%d",this->id);
		printf("\nName :%s",this->name);
		printf("\nSalary :%f",this->salary);
		printf("\nCommision :%lf\n",this->commission);
	}
	void add(){    // addition function
	 int C=this->salary + this->commission;
		printf("Addition is :%d",C);
	}
};
int main(){
	HR h1;
	h1.display();
	h1.add();

	HR h2;
	h2.setId(2);
	h2.setName("Riya");
	h2.setSalary(50.34);
	h2.setCommission(150.30);
	h2.display();
	h2.add();

	HR h3;
	h3.display();
```

```
        h3.add();


        return 0;
}
```

Q 5:

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

struct Product {

                int id;
                char pName[20];
                double price;
                float quantity ;
        //Default Constructor
        Product(){
                printf("Default Constructor !\n");
                id=0;
                strcpy(pName,"Not Assigned");
                price=0;
                quantity=0;
        }
        // Parameterised Constructor
        Product(int id ,char *pName,double price,float quantity){
                printf("Parameterised Constructor !\n");
                this->id=id;
                strcpy(this->pName,pName);
                this->price=price;
                this->quantity=quantity;
        }
        // Here setter
        void setId(int id){
```

```
                this->id=id;
        }
        void setName(char * pName){
                strcpy(this->pName,pName);
        }
        void setPrice(double price){
                this->price=price;
        }
        void setQuantity(double quantity){
                this->quantity=quantity;
        }
        // Here Display Function
        void display(){
                printf("\nId :%d",this->id);
                printf("\nProduct Name :%s",this->pName);
                printf("\nPrice :%f",this->price);
                printf("\nQuantity :%lf\n\n",this->quantity);
        }
};

int main(){
        Product p1;
        p1.display();

        Product p2(101,"Rahul",120.50,5.5);
        p2.display();
        return 0;
}
```

Q 6 :

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

```
struct salesManager {

        int id;
        char pName[20];
        double salary;
        float allowance;
    //Default Constructor
    salesManager(){
        printf("Default Constructor !");
        id=0;
        strcpy(pName,"Not Assigned");
        salary=0;
        allowance=0;
    }
    // Parameterised Constructor
    salesManager(int id ,char *pName,double salary,float allowance){
        printf("Parameterised Constructor !");
        this->id=id;
        strcpy(this->pName,pName);
        this->salary=salary;
        this->allowance=allowance;
    }
    // Here setter
    void setId(int id){
        this->id=id;
    }
    void setName(char * pName){
        strcpy(this->pName,pName);
    }
    void setPrice(double salary){
        this->salary=salary;
    }
    void setQuantity(double allowance){
```

```
                this->allowance=allowance;
        }
        // Here Display Function
        void display(){
                printf("\nId :%d",this->id);
                printf("\nProduct Name :%s",this->pName);
                printf("\nSalary :%f",this->salary);
                printf("\nAllowance :%lf\n\n",this->allowance);
        }
};

int main(){
        salesManager s1;
        s1.display();

        salesManager s2(101,"Rahul",120.50,500.5);
        s2.display();
        return 0;
}
```

Q 7 :

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

struct Student {

                int id;
                char pName[20];
                int marks;
        //Default Constructor
        Student(){
                printf("Default Constructor !");
                id=0;
```

```
            strcpy(pName,"Not Assigned");
            marks=0;
        }
        // Parameterised Constructor
        Student(int id ,char *pName,int marks){
            printf("Parameterised Constructor !");
            this->id=id;
            strcpy(this->pName,pName);
            this->marks=marks;
        }
        // Here setter
        void setId(int id){
            this->id=id;
        }
        void setName(char * pName){
            strcpy(this->pName,pName);
        }
        void setMarks(int marks){
            this->marks=marks;
        }
        // Here Display Function
        void display(){
            printf("\nId :%d",this->id);
            printf("\nProduct Name :%s",this->pName);
            printf("\nMarks %d\n\n",this->marks);
        }
};

int main(){
        Student s1;
        s1.display();

        Student s2(101,"Rahul",90);
```

```
	s2.display();
	return 0;
}
```

Q 8 :

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

struct Date
{
	int date;
	int month;
	int year;

	Date(){
		printf("\nDefault Constructor !");
		this->date=0;
		this->month=0;
		this->year=0;
	}
	Date(int date,int month,int year ){
		printf("\nParameterised Constructor !");
		this->date=date;
		this->month=month;
		this->year=year;
	}
	//setter
	void setDate(int date){
		this->date=date;
	}
	void setMonth(int month){
		this->month=month;
	}
```

```
	void setYear(int year){
		this->year=year;
	}
	//void Display
	void display(){
		printf("\nOutput : %d/%d/%d",this->date,this->month,this->year);
	}
};

int main(){
	Date d1,d2(13,04,2003),d3;
	d1.display();

	d2.display();

	d3.display();

return 0;
}
```

Q 10 :

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

struct Time{
	int hr;
	int min;
	int sec;

	Time(){
		printf("\nDefault Constructor !");
		this->hr=0;
		this->min=0;
```

```
		this->sec=0;
	}
	Time(int hr,int min,int sec ){
		printf("\nParameterised Constructor !");
		this->hr=hr;
		this->min=min;
		this->sec=sec;
	}
	//setter
	void setDate(int hr){
		this->hr=hr;
	}
	void setMonth(int min){
		this->min=min;
	}
	void setYear(int sec){
		this->sec=sec;
	}
	//void Display
	void display(){
		printf("\nOutput : %d:%d:%d",this->hr,this->min,this->sec);
	}
};

int main(){
	Time d1,d2(13,04,12),d3;
	d1.display();

	d2.display();

	d3.display();

return 0;
```

```
}
```