Cpp-Assignment 1

```
//admin
//#include<stdio.h>
#include<string.h>
#include<iostream>
using namespace std;
struct admin{
	int id;
	char name[20];
	//char* name;
	double salary;
	double allowance;

	void setId(int a){
		this->id=a;
	}
	void setName(char* nm){
		strcpy(this->name,nm);
		//name=nm; when name is the pointer not char array can assign address to address
	}
	void setSalary(double s){
		this->salary=s;
	}
	void SetAllowance(double al){
		this->allowance=al;
	}

	void display(){
		cout<<"Id:"<<this->id<<"\n";
		cout<<"name:"<<this->name<<"\n";
		cout<<"Salary:"<<this->salary<<"\n";
	cout<<"allowance:"<<this->allowance<<"\n";
}

//getters
int getid(){
	return this->id;
}

char* getname(){
	return this->name;
}

double getsalary(){
	return this->salary;
}

double getallowance(){
	return this->allowance;
}

//constructor
admin(){
	this->id=0;
	strcpy(this->name,"Not Given\n");
	this->salary=0;
	this->allowance=0;
	cout<<"Constructor called\n";
}
//parameterised constructor
admin(int i, char* nm,double s,double a){
	this->id=i;
```

```
                strcpy(this->name,nm);

                this->salary=s;

                this->allowance=a;

                cout<<"Parameterised constructor called\n";

        }

};

int main(){

        admin a1;

        printf("a1\n");

        a1.display();


        admin a2(105,"shreya",4500,789);

        a2.display();

        int a;

        char name[20];

//char* name;-->not allowed size must be define

        double salary,allowance;


        cout<<"Enter Id:";

        cin>>a;


        fflush(stdin);

        cout<<"Enter name:";

        gets(name);


        cout<<"Enter Salary:";

        cin>>salary;


        cout<<"Enter allowance:";

        cin>>allowance;
        a1.setId(a);
        a1.setName(name);
        a1.setSalary(salary);
        a1.SetAllowance(allowance);
        //a1.display();
        cout<<"After setting values of a1 by setters:\n";
        cout<<"by getters print:\n";
        cout<<"Id:"<<a1.getid()<<"\n";
        cout<<"name:"<<a1.getname()<<"\n";
        cout<<"Salary:"<<a1.getsalary()<<"\n";
        cout<<"Allowance:"<<a1.getallowance()<<"\n";
        return 0;
}
```

```
#include<stdio.h>
#include<string.h>
struct student{
        int roll_no;
        char name[20];
        double marks;

        //setters
        void setRollNo(int r){
                this->roll_no=r;
        }
        void setname(char* nm){
                strcpy(this->name,nm);
        }
        void setMarks(double m){
                this->marks=m;
        }
```

```
    void display(){
//          printf("Roll No :%d\n",this->roll_no);
//          printf("Name :%s\n",this->name);
//          printf("Marks :%lf\n",this->marks);

    printf("Roll No :%d\n",this->getrollno());
        printf("Name :%s\n",this->getname());
        printf("Marks :%lf\n",this->getMarks());
    }

    //getters
    int getrollno(){
        return this->roll_no;
    }

    char* getname(){
        return this->name;
    }

    double getMarks(){
        return this->marks;
    }

    //constructor
    student(){
        this->roll_no=0;
        strcpy(this->name,"not given");
        this->marks=0;
        printf("Constructor called\n");
    }
    //parameterised constructor
```

```
		student(int r,char *nm,double m){
			this->roll_no=r;
			strcpy(this->name,nm);
			this->marks=m;
			printf("Parameterised constructor called\n");
		}

};
int main(){
		student s1;
		printf("s1\n");
		s1.display();
		student s2(3,"prachi",56);
		s2.display();
		s1.setRollNo(1);//setRollNo(&s1,1)
		s1.setname("Prachiti");//setname(&s1,"Prachiti")
		s1.setMarks(4500);//setMarks(&s1,4500);
		//s1.display();//display(&s1);
		//(by getters)
		printf("After setting values of s1 from setters:\n");

		printf("BY GETTERS\n");
   printf("roll_NO:%d\n",s1.getrollno());
   printf("name:%s\n",s1.getname());
   printf("Marks:%.2lf\n",s1.getMarks());

		return 0;
}//#include<stdio.h>
#include<iostream>
using namespace std;
struct Date{
```

```
int day;
int mon;
int year;

//setter
void setday(int d){
		this->day=d;
}

void setmon(int m){
		this->mon=m;
}
void setyear(int y){
		this->year=y;
}

void display(){
		cout<<this->day<<"-"<<this->mon<<"-"<<this->year<<"\n";
		}
		//getters
int getDay(){
		return this->day;
}

int getMon(){
		return this->mon;
}

int getyear(){
		return this->year;
}
```

```
    //constructor
    Date(){
        this->day=0;
        this->mon=0;
        this->year=0;
        cout<<"Constructor called\n";//POC
    }

    //parameterised constructor
    Date(int d,int m,int y){
        this->day=d;
        this->mon=m;
        this->year=y;
        cout<<"Parameterised constructor called\n";
    }
};
int main(){
    Date d1;
    cout<<"d1:";
    d1.display();
    Date d2(20,6,2024);
    d2.display();
    d1.setday(10);
    d1.setmon(2);
    d1.setyear(2024);
    //d1.display();
cout<<"After setting values of d1 from setters:\n";
    cout<<"print by using getters:\n";
    cout<<d1.getDay()<<"-"<<d1.getMon()<<"-"<<d1.getyear()<<"\n";
    return 0;
```

```
}//distance
//#include<stdio.h>
#include<string.h>
#include<iostream>
using namespace std;
struct Distance{
        double feets;
        double inches;


        //setters
                void setFeets(double F){
                this->feets=F;
        }



        void setInches(double inches){
                this->inches=inches;

        }



        void display(){
                cout<<this->inches<< "inches,"<<this->feets<<"\n";
        }

        //getters

        double getfeets(){
                return this->feets;
        }
```

```
    double getinches(){
        return this->inches;
    }

    //constructor
    Distance(){
        this->feets=0;
        this->inches=0;
        cout<<"Constructor called\n";//POC
    }

    Distance(float f,float i){
        this->feets=f;
        this->inches=i;
        cout<<"Parameterised constructor called\n";
    }
};
int main(){
    Distance d;
    d.display();
    Distance d1(10,20);
    d1.display();
    d.setFeets(11.7);
    d.setInches(10.5);
    //d.display();
    cout<<"After setting values:";
    cout<<d.getfeets()<<"Feets ,"<<d.getinches()<<"inches";

    return 0;
}
```

```
//employee
//#include<stdio.h>
#include<string.h>
#include<iostream>
using namespace std;
struct employee{
        int id;
        char name[20];
        double salary;
        //this == represents address of current invoke object
        //setters
        void setid(int a){
                this->id=a;
        }
        void setname(char* nm){
                strcpy(this->name,nm);
        }
        void setsalary(double s){
                this->salary=s;
        }
        void display(){
                cout<<"id:"<<this->id<<"\n";
                cout<<"Name :"<<this->name<<"\n";
                cout<<"Salary:"<<this->salary<<"\n";

        }

        //getters
        int getid(){
                return this->id;
        }
```

```
		char* getname(){

			return this->name;
		}

		double getsalary(){
			return this->salary;
		}

		//constructor
		employee(){
			this->id=0;
			strcpy(this->name,"not given");
			this->salary=0;
			cout<<"Constructor called\n";
		}
		//parameterised constructor
		employee(int i,char* nm,double s){
			this->id=i;
			strcpy(this->name,nm);
			this->salary=s;
			cout<<"Parameterised constructor called\n";
		}
};
int main(){
		employee e1;
		printf("e1\n");
		e1.display();
		employee e2(102,"Prachiti",85000);
		e2.display();
```

```
    e1.setid(101);
    e1.setname("Hrutu");
    e1.setsalary(20000);
    //e1.display();
    cout<<"after setting values of e1 by setter :\n";
    cout<<"ID:"<<e1.getid()<<"\n";
    cout<<"Name:"<<e1.getname()<<"\n";
    cout<<"Salary:"<<e1.getsalary()<<"\n";


    return 0;
}
```

```
//hr
//#include<stdio.h>
#include<string.h>
#include<iostream>
using namespace std;
struct Hr{
    int id;
    char name[20];
    double salary;
    double commission;

    void setId(int a){
        this->id=a;
    }
    void setName(const char* nm){
        strcpy(this->name,nm);
    }
    void setSalary(double s){
        this->salary=s;
    }
    void setCommission(double c){
        this->commission=c;
    }
    void display(){
        cout<<"Id:"<<this->id<<"\n";
        cout<<"name:"<<this->name<<"\n";
        cout<<"Salary:"<<this->salary<<"\n";
        cout<<"commission:"<<this->commission<<"\n";
    }
    //getters
    int getid(){
        return this->id;
    }
    char* getname(){
        return this->name;
    }

    double getsalary(){
        return this->salary;
    }

    double getcommision(){
        return this->commission;
    }

    //constructor
    Hr(){
        this->id=0;
        strcpy(this->name,"Not Given");
        this->salary=0;
        this->commission=0;
        cout<<"Constructor called\n";
    }
    //parameterised constructor
    Hr(int i,char* nm,double s,double c){
        this->id=i;
        strcpy(this->name,nm);
        this->salary=s;
        this->commission=c;
        cout<<"Parameterised constructor called\n";
    }
};
```

```
int main(){
	Hr h1;
	printf("h1\n");
	h1.display();
	Hr h2(104,"pravin",5600,780);
	h2.display();
	h1.setId(103);
	// ISO C++ forbids converting a string constant to 'char*' [-Wwrite-strings]-->const required when we pass the  hardcoded string as paramter
	h1.setName("prachiti");//here "prachiti" is store in dummy so in defination when we catch the address that point varible shouble be constant "Prachiti" pointer jyala point kart ahe ti value constant asli pahije //complier saying that dont change the address
	h1.setSalary(200000);
	h1.setCommission(5000);
	//h1.display();
	cout<<"After Setting the value of h1 from the setters:\n";
cout<<"print by using the getters:\n";

	cout<<"ID:"<<h1.getid()<<"\n";
	cout<<"Name:"<<h1.getname()<<"\n";
	cout<<"salary:"<<h1.getsalary()<<"\n";
	cout<<"Commission:"<<h1.getcommision()<<"\n";
	return 0;
}
//product
//#include<stdio.h>
#include<string.h>
#include<iostream>
using namespace std;
struct Product {
	int id;
```

```
char name[20];
int quantity;
double price;

//setters
void setId(int id){
        this->id=id;
}
void setName(char* nm){
        strcpy(this->name,nm);
}
void setQuantity(int q){
        this->quantity=q;
}
void setPrice(double p){
        this->price=p;
}

//getters
int getId(){
        return this->id;
}

char* getName(){
        return this->name;
}

int getQuantity(){
        return this->quantity;
}
```

```
		double getPrice(){
			return this->price;
		}

		void display(){
			cout<<"Id:"<<this->id<<"\n";
			cout<<"product name "<<this->name<<"\n";
			cout<<"Quantity:%d\n",this->quantity<<"\n";
			cout<<"Price:%lf\n",this->price<<"\n";
		}

		//constructor
		//defalut constructor
		Product(){
			this->id=0;
			strcpy(this->name,"Product");
			this->quantity=0;
			this->price=0;
			cout<<"Defalut constructor called\n";
		}

		//parameterised constructor
		Product(int i, char* nm, int q ,double p){
			this->id=i;
			strcpy(this->name,nm);
			this->quantity=q;
			this->price=p;
			cout<<"Parameterised constructor called\n";
		}

	};
```

```
int main(){
	Product p;
	printf("P\n");
	p.display();
	Product p1(1,"Book",56,908);
	p1.display();

	p.setId(3);
	p.setName("Soap");
	p.setQuantity(9);
	p.setPrice(80);

	cout<<"After setting the values of p from setters\n by getter printing:\n";
	cout<<"id:%d\n",p.getId()<<"\n";
	cout<<"Product_name:%s\n",p.getName());
	printf("Quantity:%d\n",p.getQuantity());
	printf("Price:%lf\n",p.getPrice());
	return 0;
}
#include<stdio.h>

struct time{
	int hr;
	int min;
	int sec;

	//setters
	void setHr(int h){
		this->hr=h;
	}
```

```
void setMin(int m){
	this->min=m;
}
void setSec(int s){
	this->sec=s;
}

void display(){
	printf("%d:%d:%d\n",this->hr,this->min,this->sec);
}

//getters
int getHr(){
	return this->hr;
}

int getMin(){
	return this->min;
}

int getsec(){
	return this->sec;
}

//constructor
time(){
	this->hr=0;
	this->min=0;
	this->sec=0;
	printf("Constructor called\n");//POC
}
```

```
    time(int h, int m, int s){

        this->hr=h;

        this->min=m;

        this->sec=s;

        printf("Parameterised constructor called\n");

    }

};

int main(){

    time t1;

    printf("t1\n");

    t1.display();

    time t2(2,23,59);

    t2.display();

    t1.setHr(10);

    t1.setMin(20);

    t1.setSec(50);

    //t1.display();

    printf("After setting values of t1 by setters:");

    printf("%d:%d:%d",t1.getHr(),t1.getMin(),t1.getsec());

    return 0;

}
```