

Assignment 01

1)

```
#include <iostream>
using namespace std;
#include <string.h>
typedef struct Student
{
    int rollNo;
    char name[30];
    float marks;
    Student()
    {
        cout << "\nDefault Constructor called";
        strcpy(this->name, "Not Given");
        this->rollNo = 0;
        this->marks = 0.00;
    }
public:
    Student(char *name, int rollNo, int marks)
    {
        cout << "\nParametreized Constructor called";
        strcpy(this->name, name);
        this->rollNo = rollNo;
        this->marks = marks;
    }
    void setRollNo(int rollNo)
    {
        this->rollNo = rollNo;
    }
    void setName(char *name)
    {
        strcpy(this->name, name);
    }
    void setMarks(float marks)
    {
        this->marks = marks;
    }

    int getRollNo()
    {
        return this->rollNo;
    }
    char *getName()
    {
        return this->name;
    }
    float getMarks()
    {
        return this->marks;
    }
}
```

```

}

void display()
{
    cout << "\nRoll No:" << this->rollNo;
    cout << "\nName :" << this->name;
    cout << "\nMarks :" << this->marks;
}
} Student;

int main()
{
    Student s1, s2("Bhagvat", 96, 98);
    int rn;
    float marks;
    char name[20];

    // Called After Constructor
    cout << "\nDefault Values Display";
    s1.display();
    s2.display();
    // Normal After Setters

    cout << "\n\nUsing Setters s1";
    cout << "\nEnter Student Roll No :";
    cin >> rn;
    s1.setRollNo(rn);

    cout << "\nEnter Student Name :";
    cin >> name;
    s1.setName(name);

    cout << "\nEnter Student Marks :";
    cin >> marks;
    s1.setMarks(marks);

    cout << "\n s1 Display";
    s1.display();

    // cout << "\n\nUsing Setters s2";
    // cout << "\nEnter Student Roll No :";
    // cin >> rn;
    // s2.setRollNo(rn);

    // cout << "\nEnter Student Name :";
    // cin >> name;
    // s2.setName(name);

    // cout << "\nEnter Student Marks :";
    // cin >> marks;
    // s2.setMarks(marks);

    // cout << "\n s2 Display";

```

```

    // s2.display();
    cout << "\n\nGetters s1";
    cout << "\nRoll No:" << s1.getRollNo() << "\tName :" << s1.getName() << "\tMarks :" <<
s1.getMarks();
    // cout << "\n\nGetters s2";
    // cout << "\nRoll No:" << s2.getRollNo() << "\tName :" << s2.getName() << "\tMarks :"
<< s2.getMarks();

    return 0;
}

```

Output:

PS D:\Fullstack-Java-FirstBit-Solutions\Basic-C-and-CPP\CPP\Assignments\Assignment01\output> &
.\'struct01Student.exe'

Default Constructor called

Parametreized Constructor called

Default Values Display

Roll No:0

Name :Not Given

Marks :0

Roll No:96

Name :Bhagvat

Marks :98

Using Setters s1

Enter Student Roll No :34

Enter Student Name :shdfis

Enter Student Marks :431

s1 Display

Roll No:34

Name :shdfis

Marks :431

Getters s1

Roll No:34 Name :shdfis Marks :431

PS D:\Fullstack-Java-FirstBit-Solutions\Basic-C-and-CPP\CPP\Assignments\Assignment01\output>

2)

```
#include <iostream>
#include <string.h>
using namespace std;
typedef struct Employee
{
    int id;
    char name[20];
    float salary;
    // Constructor
    Employee()
    {
        cout << "\nDefault constructor called\n";
        this->id = NULL;
        strcpy(this->name, "No Name");
        this->salary = NULL;
    }

    Employee(char *name, int id, float salary)
    {
        cout << "\nParameterized Constructor for Employee called";
        strcpy(this->name, name);
        this->id = id;
        this->salary = salary;
    }
public:
    // Setters
    void setId(int Id)
    {
        this->id = Id;
    }
    void setName(char *name)
    {
        strcpy(this->name, name);
    }
    void setSalary(float salary)
    {
        this->salary = salary;
    }
    // Getters
    int getId()
    {
        return this->id;
    }
    char *getName()
    {
        return this->name;
    }
    float getSalary()
    {
        return this->salary;
    }
}
```

```

    }
    // Display
    void dispaly()
    {
        cout << "\nId : " << this->id << "\tName :" << this->name << "\t Salary :" <<
this->salary;
    }
} Employee;

int main()
{
    int id;
    char name[20];
    float salary;
    Employee e1, e2("Bhagvat", 96, 750000);
    // Constructor call
    cout << "\nDefault Values of E1 :";
    e1.dispaly();
    cout << "\nDefault Values of E2 :";
    e2.dispaly();
    // E1 Setters
    cout << "\nSetters For e1";
    cout << "\nEnter ID :";
    cin >> id;
    e1.setId(id);
    cout << "\nEnter Name :";
    cin >> name;
    e1.setName(name);
    cout << "\nEnter Salary :";
    cin >> salary;
    e1.setSalary(salary);
    // Display
    cout << "\nDisaplay e1";
    e1.dispaly();

    // // E2 Setters
    // cout << "\nSetters For e2";
    // cout << "\nEnter ID :";
    // cin >> id;
    // e2.setId(id);
    // cout << "\nEnter Name :";
    // cin >> name;
    // e2.setName(name);
    // cout << "\nEnter Salary :";
    // cin >> salary;
    // e2.setSalary(salary);
    // // Display
    // cout << "\nDisaplay e1";
    // e2.dispaly();

    // E1 Getters
    cout << "\nGetters E1";

```

```

    cout << "\nId : " << e1.getId() << "\tName :" << e1.getName() << "\t Salary :" <<
e1.getSalary();
    // // E2 Getters
    // cout << "\nGetters E2";
    // cout << "\nId : " << e2.getId() << "\tName :" << e2.getName() << "\t Salary :" <<
e2.getSalary();
    return 1;
}

```

Output:

PS D:\Fullstack-Java-FirstBit-Solutions\Basic-C-and-CPP\CPP\Assignments\Assignment01\output> &
.\'struct02Employee.exe'

Default constructor called

Parameterized Constructor for Employee called

Default Values of E1 :

Id : 0 Name :No Name Salary :0

Default Values of E2 :

Id : 96 Name :Bhagvat Salary :750000

Setters For e1

Enter ID :123

Enter Name :jHVwdi

Enter Salary :2342

Display e1

Id : 123 Name :jHVwdi Salary :2342

Getters E1

Id : 123 Name :jHVwdi Salary :2342

PS D:\Fullstack-Java-FirstBit-Solutions\Basic-C-and-CPP\CPP\Assignments\Assignment01\output>

3)

```
#include <iostream>
#include <string.h>
using namespace std;
typedef struct Admin
{
    // id,name,salary,allowence
    int id;
    char name[20];
    float salary;
    float allowance;

    // Construuctor
    Admin()
    {
        cout << "\nDefault constructor called\n";
        this->id = 00;
        strcpy(this->name, "NoName");
        this->salary = 00;
        this->allowence = 00;
    }

    Admin(char *name, int id, float salary, float allowance)
    {
        cout << "\nParameterized Constructor for Admin called";
        strcpy(this->name, name);
        this->id = id;
        this->salary = salary;
        this->allowence = allowance;
    }
}

public:
    // Setters
    void setId(int id) { this->id = id; }
    void setName(char name[]) { strcpy(this->name, name); }
    void setSalary(float salary) { this->salary = salary; }
    void setAllowence(float allowance) { this->allowence = allowance; }

    // Display
    void display()
    {
        cout << "\n\nId :" << this->id << "\tName :" << this->name << "\tSalary" << this->salary << "\tAllowence :" << this->allowence;
    }

    // getters
    int getId() { return this->id; }
    char *getName() { return this->name; }
    float getSalary() { return this->salary; }
    float getAllowence() { return this->allowence; }
} Admin;
```

```

int main()
{
    Admin Admin1, Admin2("Bhagvat", 96, 75599999, 230000);
    int id;
    char name[20];
    float salary, allowance;

    // Constructor call
    cout << "\nDefault Admin1 values:";
    Admin1.display();
    // Constructor call
    cout << "\nDefault Admin2 values:";
    Admin2.display();

    // Setters for Admin
    cout << "\nEnter Admin ID: ";
    cin >> id;
    Admin1.setId(id);

    cout << "Enter Admin Name: ";
    cin >> name;
    Admin1.setName(name);

    cout << "Enter Admin Salary: ";
    cin >> salary;
    Admin1.setSalary(salary);

    cout << "Enter Admin allowance: ";
    cin >> allowance;
    Admin1.setAllowance(allowance);

    cout << "\nAdmin1 Display";
    // Display Admin data
    Admin1.display();

    // // Getters
    // cout << "\nGetters Admin1";
    // cout << "\nId : " << Admin1.getId() << "\tName : " << Admin1.getName() << "\tSalary
: " << Admin1.getSalary() << "\tallowance : " << Admin1.getAllowance();

    // // Setters for Admin
    // cout << "\nEnter Admin ID: ";
    // cin >> id;
    // Admin2.setId(id);

    // cout << "Enter Admin Name: ";
    // cin >> name;
    // Admin2.setName(name);

    // cout << "Enter Admin Salary: ";
    // cin >> salary;
    // Admin2.setSalary(salary);

```



```

    // cout << "Enter Admin allowance: ";
    // cin >> allowance;
    // Admin2.setAllowence(allowence);
    // // Getters Admin2
    // cout << "\nGetters Admin2";
    // cout << "\nId : " << Admin2.getId() << "\tName : " << Admin2.getName() << "\tSalary
: " << Admin2.getSalary() << "\tallowence : " << Admin2.getAllowence();
    // // Display Admin data
    // cout << "\n\nAdmin2 Display";
    // Admin2.display();

    return 1;
}

```

Output:

PS D:\Fullstack-Java-FirstBit-Solutions\Basic-C-and-CPP\CPP\Assignments\Assignment01\output> &
.\'struct03Admin.exe'

Default constructor called

Parameterized Constructor for Admin called

Default Admin1 values:

Id :0 Name :NoName Salary0 Allowence :0

Default Admin2 values:

Id :96 Name :Bhagvat Salary7.56e+07 Allowence :230000

Enter Admin ID: 234

Enter Admin Name: Ajjuf

Enter Admin Salary: 3125

Enter Admin allowance: 433

Admin1 Display

Id :234 Name :Ajjuf Salary3125 Allowence :433

PS D:\Fullstack-Java-FirstBit-Solutions\Basic-C-and-CPP\CPP\Assignments\Assignment01\output>

4)

```
#include <iostream>
#include <string.h>
using namespace std;

typedef struct HR
{
    int id;
    char name[20];
    float salary;
    float commission;

    // Constructor
    HR()
    {
        cout << "\nDefault constructor called\n";
        this->id = 0;
        strcpy(this->name, "No Name");
        this->salary = 0;
        this->commission = 0;
    }
    HR(char *name, int id, float salary, float commission)
    {
        cout << "\nParameterized Constructor for HR called";
        strcpy(this->name, name);
        this->id = id;
        this->salary = salary;
        this->commission = commission;
    }

public:
    // Setters
    void setId(int Id) { this->id = Id; }
    void setName(char *name) { strcpy(this->name, name); }
    void setSalary(float salary) { this->salary = salary; }
    void setCommission(float commission) { this->commission = commission; }

    // Getters
    int getId() { return this->id; }
    char *getName() { return this->name; }
    float getSalary() { return this->salary; }
    float getCommission() { return this->commission; }

    // Display
    void display()
    {
        cout << "\nId : " << this->id << "\tName : " << this->name << "\tSalary : " <<
this->salary << "\tCommission : " << this->commission;
    }
} HR;
```

```

int main()
{
    HR hr1, hr2("Bhagvat", 231, 435332, 2324);
    int id;
    char name[20];
    float salary, commission;

    // Constructor call
    cout << "\nDefault HR1 values:";
    hr1.display();
    // Constructor call
    cout << "\nDefault HR2 values:";
    hr2.display();

    // Setters for HR
    cout << "\nEnter HR ID: ";
    cin >> id;
    hr1.setID(id);

    cout << "Enter HR Name: ";
    cin >> name;
    hr1.setName(name);

    cout << "Enter HR Salary: ";
    cin >> salary;
    hr1.setSalary(salary);

    cout << "Enter HR Commission: ";
    cin >> commission;
    hr1.setCommission(commission);

    // Display HR data
    hr1.display();

    // Getters
    // cout << "\nGetters HR1";
    // cout << "\nId : " << hr1.getId() << "\tName : " << hr1.getName() << "\tSalary : "
    << hr1.getSalary() << "\tCommission: " << hr1.getCommission();

    // // Setters for HR
    // cout << "\nEnter HR ID: ";
    // cin >> id;
    // hr2.setID(id);

    // cout << "Enter HR Name: ";
    // cin >> name;
    // hr2.setName(name);

    // cout << "Enter HR Salary: ";
    // cin >> salary;
    // hr2.setSalary(salary);

    // cout << "Enter HR Commission: ";

```

```

    // cin >> commission;
    // hr2.setCommission(commission);
    // // Getters hr2
    // cout << "\nGetters HR2";
    // cout << "\nId : " << hr2.getId() << "\tName : " << hr2.getName() << "\tSalary : "
    << hr2.getSalary() << "\tCommission : " << hr2.getCommission();
    // // Display HR data
    // hr2.display();

    return 1;
}

```

Output:

```

PS D:\Fullstack-Java-FirstBit-Solutions\Basic-C-and-CPP\CPP\Assignments\Assignment01\output> &
.\'struct04HR.exe'

```

Default constructor called

Parameterized Constructor for HR called

Default HR1 values:

Id : 0 Name : No Name Salary : 0 Commission : 0

Default HR2 values:

Id : 231 Name : Bhagvat Salary : 435332 Commission : 2324

Enter HR ID: 123

Enter HR Name: abfiyearu

Enter HR Salary: 3241

Enter HR Commission: 34

Id : 123 Name : abfiyearu Salary : 3241 Commission : 34

```

PS D:\Fullstack-Java-FirstBit-Solutions\Basic-C-and-CPP\CPP\Assignments\Assignment01\output>

```

5)

```

#include <iostream>
#include <string.h>
using namespace std;

typedef struct SalesManager
{
    int id;
    char name[20];
    float salary;
}

```

```

float incentive;
int target;

// Constructor
SalesManager()
{
    cout << "\nDefault constructor called\n";
    this->id = NULL;
    strcpy(this->name, "No Name");
    this->salary = NULL;
    this->incentive = NULL;
    this->target = NULL;
}

SalesManager(char *name, int id, float salary, float incentive, int target)
{
    cout << "\nParameterized Constructor for SalesManager called";
    strcpy(this->name, name);
    this->id = id;
    this->salary = salary;
    this->incentive = incentive;
    this->target = target;
}

public:
    // Setters
    void setId(int Id) { this->id = Id; }
    void setName(char *name) { strcpy(this->name, name); }
    void setSalary(float salary) { this->salary = salary; }
    void setIncentive(float incentive) { this->incentive = incentive; }
    void setTarget(int target) { this->target = target; }

    // Getters
    int getId() { return this->id; }
    char *getName() { return this->name; }
    float getSalary() { return this->salary; }
    float getIncentive() { return this->incentive; }
    int getTarget() { return this->target; }

    // Display
    void display()
    {
        cout << "\nId : " << this->id << "\tName : " << this->name
            << "\tSalary : " << this->salary << "\tIncentive : " << this->incentive
            << "\tTarget : " << this->target;
    }
} SalesManager;

int main()
{
    SalesManager sm1, sm2("Bhagvat", 123, 23123, 432, 21);
    int id, target;

```

```

char name[20];
float salary, incentive;

// Constructor call
cout << "\nDefault SalesManager1 values:";
sm1.display();
// Constructor call
cout << "\nDefault SalesManager2 values:";
sm2.display();

// Setters for SalesManager1
cout << "\nEnter SalesManager ID: ";
cin >> id;
sm1.setId(id);

cout << "Enter SalesManager Name: ";
cin >> name;
sm1.setName(name);

cout << "Enter SalesManager Salary: ";
cin >> salary;
sm1.setSalary(salary);

cout << "Enter SalesManager Incentive: ";
cin >> incentive;
sm1.setIncentive(incentive);

cout << "Enter SalesManager Target: ";
cin >> target;
sm1.setTarget(target);

// Display SalesManager1 data
sm1.display();

// Getters for SalesManager1
// cout << "\nGetters SalesManager1";
// cout << "\nId : " << sm1.getId() << "\tName : " << sm1.getName() << "\tSalary : "
<< sm1.getSalary() << "\tIncentive : " << sm1.getIncentive() << "\tTarget : " <<
sm1.getTarget();

// // Setters for SalesManager2
// cout << "\nEnter SalesManager ID: ";
// cin >> id;
// sm2.setId(id);

// cout << "Enter SalesManager Name: ";
// cin >> name;
// sm2.setName(name);

// cout << "Enter SalesManager Salary: ";
// cin >> salary;
// sm2.setSalary(salary);

```

```

// cout << "Enter SalesManager Incentive: ";
// cin >> incentive;
// sm2.setIncentive(incentive);

// cout << "Enter SalesManager Target: ";
// cin >> target;
// sm2.setTarget(target);

// // Getters SalesManager2
// cout << "\nGetters SalesManager2";
// cout << "\nId : " << sm2.getId() << "\tName : " << sm2.getName() << "\tSalary : "
<< sm2.getSalary() << "\tIncentive : " << sm2.getIncentive() << "\tTarget : " <<
sm2.getTarget();

// // Display SalesManager2 data
// sm2.display();

return 1;
}

```

Output:

PS D:\Fullstack-Java-FirstBit-Solutions\Basic-C-and-CPP\CPP\Assignments\Assignment01\output> &
 .\struct05SalesManager.exe'

Default constructor called

Parameterized Constructor for SalesManager called

Default SalesManager1 values:

Id : 0 Name : No Name Salary : 0 Incentive : 0 Target : 0

Default SalesManager2 values:

Id : 123 Name : Bhagvat Salary : 23123 Incentive : 432 Target : 21

Enter SalesManager ID: 123

Enter SalesManager Name: dskguie

Enter SalesManager Salary: 84736

Enter SalesManager Incentive: 4892

Enter SalesManager Target: 3

Id : 123 Name : dskguie Salary : 84736 Incentive : 4892 Target : 3

PS D:\Fullstack-Java-FirstBit-Solutions\Basic-C-and-CPP\CPP\Assignments\Assignment01\output>

6)

```
#include <iostream>
#include <string.h>
using namespace std;

typedef struct Date
{
    int day;
    int month;
    int year;

    // Constructor
    Date()
    {
        cout << "\nDefault constructor called\n";
        this->day = 1;
        this->month = 1;
        this->year = 2000;
    }

    Date(int day, int month, int year)
    {
        cout << "\nParameterized Constructor for Date called";
        this->day = day;
        this->month = month;
        this->year = year;
    }
}

public:
    // Setters
    void setDay(int d) { this->day = d; }
    void setMonth(int m) { this->month = m; }
    void setYear(int y) { this->year = y; }

    // Getters
    int getDay() { return this->day; }
    int getMonth() { return this->month; }
    int getYear() { return this->year; }

    // Display
    void display()
    {
        cout << "\nDate: " << this->day << "/" << this->month << "/" << this->year;
    }
} Date;

int main()
{
    Date date1, date2(11, 10, 2002);
    int day, month, year;
```



```

// Constructor call
cout << "\nDefault Date1 values:";
date1.display();
// Constructor call
cout << "\nDefault Date2 values:";
date2.display();

// Setters for Date1
cout << "\nEnter Day: ";
cin >> day;
date1.setDay(day);

cout << "Enter Month: ";
cin >> month;
date1.setMonth(month);

cout << "Enter Year: ";
cin >> year;
date1.setYear(year);

// Display Date1 data
date1.display();

// Getters for Date1
cout << "\nGetters Date1";
cout << "\nDay : " << date1.getDay() << "\tMonth : " << date1.getMonth() << "\tYear : " << date1.getYear();

// // Setters for Date2
// cout << "\nEnter Day: ";
// cin >> day;
// date2.setDay(day);

// cout << "Enter Month: ";
// cin >> month;
// date2.setMonth(month);

// cout << "Enter Year: ";
// cin >> year;
// date2.setYear(year);

// // Getters Date2
// cout << "\nGetters Date2";
// cout << "\nDay : " << date2.getDay() << "\tMonth : " << date2.getMonth() << "\tYear : " << date2.getYear();

// // Display Date2 data
// date2.display();

return 1;
}

```

Output:

```
PS D:\Fullstack-Java-FirstBit-Solutions\Basic-C-and-CPP\CPP\Assignments\Assignment01\output> &
.\'struct06Date.exe'
```

Default constructor called

Parameterized Constructor for Date called

Default Date1 values:

Date: 1/1/2000

Default Date2 values:

Date: 11/10/2002

Enter Day: 24

Enter Month: 11

Enter Year: 2002

Date: 24/11/2002

Getters Date1

Day : 24 Month : 11 Year : 2002

```
PS D:\Fullstack-Java-FirstBit-Solutions\Basic-C-and-CPP\CPP\Assignments\Assignment01\output>
```

7)

```
#include <iostream>
#include <string.h>
using namespace std;
typedef struct Time
{
    int hour;
    int min;
    int sec;

    // Constructor
    Time()
    {
        cout << "\nDefault constructor called\n";
        this->hour = 0;
        this->min = 0;
        this->sec = 0;
    }
    Time(int hour, int min, int sec)
```

```

{
    cout << "\nParameterized Constructor for Time called";
    this->hour = hour;
    this->min = min;
    this->sec = sec;
}

public:
    // Setters
    void setHour(int h) { this->hour = h; }
    void setMin(int m) { this->min = m; }
    void setSec(int s) { this->sec = s; }

    // Getters
    int getHour() { return this->hour; }
    int getMin() { return this->min; }
    int getSec() { return this->sec; }

    // Display
    void display()
    {
        cout << "\nTime: " << this->hour << ":" << this->min << ":" << this->sec;
    }
} Time;

int main()
{
    Time time1, time2(12, 32, 43);
    int hour, min, sec;

    // Constructor call
    cout << "\nDefault Time1 values:";
    time1.display();
    // Constructor call
    cout << "\nDefault Time2 values:";
    time2.display();

    // Setters for Time1
    cout << "\nEnter Hour: ";
    cin >> hour;
    time1.setHour(hour);

    cout << "Enter Minute: ";
    cin >> min;
    time1.setMin(min);

    cout << "Enter Second: ";
    cin >> sec;
    time1.setSec(sec);

    // Display Time1 data
    time1.display();
}

```

```

// Getters for Time1
cout << "\nGetters Time1";
cout << "\nHour : " << time1.getHour() << "\tMinute : " << time1.getMin() << "\tSecond
: " << time1.getSec();

// Setters for Time2
// cout << "\nEnter Hour: ";
// cin >> hour;
// time2.setHour(hour);

// cout << "Enter Minute: ";
// cin >> min;
// time2.setMin(min);

// cout << "Enter Second: ";
// cin >> sec;
// time2.setSec(sec);

// // Getters Time2
// cout << "\nGetters Time2";
// cout << "\nHour : " << time2.getHour() << "\tMinute : " << time2.getMin() <<
"\tSecond : " << time2.getSec();

// // Display Time2 data
// time2.display();

return 1;
}

```

Output:PS D:\Fullstack-Java-FirstBit-Solutions\Basic-C-and-CPP\CPP\Assignments\Assignment01\output> & .\struct07Time.exe'

Default constructor called

Parameterized Constructor for Time called

Default Time1 values:

Time: 0:0:0

Default Time2 values:

Time: 12:32:43

Enter Hour: 45

Enter Minute: 23

Enter Second: 43

Time: 45:23:43

Getters Time1

Hour : 45 Minute : 23 Second : 43

PS D:\Fullstack-Java-FirstBit-Solutions\Basic-C-and-CPP\CPP\Assignments\Assignment01\output>

8)

```
#include <iostream>
#include <string.h>
using namespace std;

typedef struct Distance
{
    int feet;
    int inch;

    // Constructor
    Distance()
    {
        cout << "\nDefault constructor called";
        this->feet = 0;
        this->inch = 0;
    }

    Distance(int feet, int inch)
    {
        cout << "\nParameterized Constructor for Distance called";
        this->feet = feet;
        this->inch = inch;
    }

public:
    // Setters
    void setFeet(int f) { this->feet = f; }
    void setInch(int i) { this->inch = i; }

    // Getters
    int getFeet() { return this->feet; }
    int getInch() { return this->inch; }

    // Display
    void display()
    {
        cout << "\nDistance: " << this->feet << " feet " << this->inch << " inches";
    }
} Distance;

int main()
{
    Distance dist1, dist2(23, 43);
    int feet, inch;

    // Constructor call
    cout << "\nDefault Distance1 values:";
    dist1.display();
    // Constructor call
    cout << "\nDefault Distance2 values:";
    dist2.display();
}
```

```

// Setters for Distance1
cout << "\nEnter Feet: ";
cin >> feet;
dist1.setFeet(feet);

cout << "Enter Inch: ";
cin >> inch;
dist1.setInch(inch);

// Display Distance1 data
dist1.display();

// Getters for Distance1
cout << "\nGetters Distance1";
cout << "\nFeet : " << dist1.getFeet() << "\tInch : " << dist1.getInch();

return 1;
}

```

Output:PS D:\Fullstack-Java-FirstBit-Solutions\Basic-C-and-CPP\CPP\Assignments\Assignment01\output> &
.\'struct08Distance.exe'

Default constructor called

Parameterized Constructor for Distance called

Default Distance1 values:

Distance: 0 feet 0 inches

Default Distance2 values:

Distance: 23 feet 43 inches

Enter Feet: 23

Enter Inch: 12

Distance: 23 feet 12 inches

Getters Distance1

Feet : 23 Inch : 12

PS D:\Fullstack-Java-FirstBit-Solutions\Basic-C-and-CPP\CPP\Assignments\Assignment01\output>

9)

```
#include <iostream>
#include <string.h>
using namespace std;

typedef struct Complex
{
    int real;
    int imaginary;

    // Constructor
    Complex()
    {
        cout << "\nDefault constructor called\n";
        this->real = 0;
        this->imaginary = 0;
    }

    Complex(int real, int imaginary)
    {
        cout << "\nParameterized Constructor for Complex called\n";
        this->real = real;
        this->imaginary = imaginary;
    }

public:
    // Setters
    void setReal(int r) { this->real = r; }
    void setImaginary(int i) { this->imaginary = i; }

    // Getters
    int getReal() { return this->real; }
    int getImaginary() { return this->imaginary; }

    // Display
    void display()
    {
        cout << "\nComplex Number: " << this->real << " + " << this->imaginary << "i";
    }

    Complex add(Complex c)
    {
        cout << "\nInside Add Function";
        Complex temp;
        temp.real = this->real + c.real;
        temp.imaginary = this->imaginary + c.imaginary;
        return temp;
    }
} Complex;

int main()
{
```

```

Complex complex1, complex2(30, 49);
int real, imaginary;

// Constructor call
cout << "\nDefault Complex1 values:";
complex1.display();
// Constructor call
cout << "\nDefault Complex2 values:";
complex2.display();

// Setters for Complex1
cout << "\nEnter Real part: ";
cin >> real;
complex1.setReal(real);

cout << "Enter Imaginary part: ";
cin >> imaginary;
complex1.setImaginary(imaginary);

// Display Complex1 data
complex1.display();

// Getters for Complex1
cout << "\nGetters Complex1";
cout << "\nReal : " << complex1.getReal() << "\tImaginary : " <<
complex1.getImaginary();

// // Setters for Complex2
// cout << "\nEnter Real part: ";
// cin >> real;
// complex2.setReal(real);

// cout << "Enter Imaginary part: ";
// cin >> imaginary;
// complex2.setImaginary(imaginary);

// // Getters Complex2
// cout << "\nGetters Complex2";
// cout << "\nReal : " << complex2.getReal() << "\tImaginary : " <<
complex2.getImaginary();

// // Display Complex2 data
// complex2.display();

// Addition
Complex complex3 = complex1.add(complex2);
cout << "Addition of C1 & C2 = ";
complex3.display();
return 1;
}

```

Output:


```
PS D:\Fullstack-Java-FirstBit-Solutions\Basic-C-and-CPP\CPP\Assignments\Assignment01\output> &
.\'struct09Complex.exe'
```

Default constructor called

Parameterized Constructor for Complex called

Default Complex1 values:

Complex Number: 0 + 0i

Default Complex2 values:

Complex Number: 30 + 49i

Enter Real part: 34

Enter Imaginary part: 123

Complex Number: 34 + 123i

Getters Complex1

Real : 34 Imaginary : 123

Inside Add Function

Default constructor called

Addition of C1 & C2 =

Complex Number: 64 + 172i

```
PS D:\Fullstack-Java-FirstBit-Solutions\Basic-C-and-CPP\CPP\Assignments\Assignment01\output>
```

10)

```
#include <iostream>
#include <string.h>
using namespace std;

struct Product
{
    int id;
    char name[20];
    int quantity;
    float price;

    // Constructor
    Product()
```

```

{
    cout << "\nDefault Constructor Called";
    this->id = NULL;
    strcpy(this->name, "No Name");
    this->quantity = 0;
    this->price = 0;
}
Product(char *name, int id, int quantity, float price)
{
    cout << "\nParameterized Constructor for Product called";
    strcpy(this->name, name);
    this->id = id;
    this->quantity = quantity;
    this->price = price;
}

public:
    // Setters
    void setId(int Id) { this->id = Id; }
    void setName(char *name) { strcpy(this->name, name); }
    void setQuantity(int qty) { this->quantity = qty; }
    void setPrice(float price) { this->price = price; }

    // Getters
    int getId() { return this->id; }
    char *getName() { return this->name; }
    int getQuantity() { return this->quantity; }
    float getPrice() { return this->price; }

    // Display
    void display()
    {
        cout << "\nProduct ID: " << this->id << "\tName: " << this->name
            << "\tQuantity: " << this->quantity << "\tPrice: " << this->price;
    }
};

int main()
{
    Product prod1, prod2("Laptop", 12, 34, 4000.0);
    int productId, quantity;
    char name[20];
    float price;

    // Constructor call
    cout << "\nDefault Product1 values:";
    prod1.display();
    // Constructor call
    cout << "\nDefault Product2 values:";
    prod2.display();

    // Setters for Product1
    cout << "\nEnter Product ID: ";
    cin >> productId;

```

```

prod1.setId(productId);

cout << "Enter Product Name: ";
cin >> name;
prod1.setName(name);

cout << "Enter Product Price: ";
cin >> price;
prod1.setPrice(price);

cout << "Enter Product Quantity: ";
cin >> quantity;
prod1.setQuantity(quantity);

// Display Product1 data
prod1.display();

// Getters for Product1
cout << "\nGetters Product1";
cout << "\nProductId : " << prod1.getId() << "\tName : " << prod1.getName() <<
"\tPrice : " << prod1.getPrice() << "\tQuantity : " << prod1.getQuantity();
return 1;
}

```

Output:

PS D:\Fullstack-Java-FirstBit-Solutions\Basic-C-and-CPP\CPP\Assignments\Assignment01\output> &
 .\struct10Product.exe'

Default Constructor Called

Parameterized Constructor for Product called

Default Product1 values:

Product ID: 0 Name: No Name Quantity: 0 Price: 0

Default Product2 values:

Product ID: 12 Name: Laptop Quantity: 34 Price: 4000

Enter Product ID: 23

Enter Product Name: sjafyakgy

Enter Product Price: 72323

Enter Product Quantity: 231

Product ID: 23 Name: sjafyakgy Quantity: 231 Price: 72323

Getters Product1

ProductId : 23 Name : sjafyakgy Price : 72323 Quantity : 231

PS D:\Fullstack-Java-FirstBit-Solutions\Basic-C-and-CPP\CPP\Assignments\Assignment01\output> p