

Object Oriented Programming Lab

Assignment 3

Submitted by:

Navdeep Singh

12th August 2025

Roll No: 24124073

Group: 3

Branch: Information Technology

Year: 2nd Year

Practice Question to practice Structures in C++ (Make a Simple struct Student with name ROLL No and Marks take input from user and display it)

Code

```
1 #include<bits/stdc++.h>
2 using namespace std;
3
4 struct Student{
5     private :
6     string name;
7     int rollNo;
8     float marks;
9
10    public:
11    void takeInputs(Student &s1){
12        cout<<"Inside take inputs Functions !!\n";
13        cout<<"Enter Name of Student : ";
14        getline(cin,s1.name);
15        cout<<"Enter Roll No of Student : ";
16        cin>>s1.rollNo;
17        cout<<"Enter Marks of Student : ";
18        cin>>s1.marks;
19    }
20
21    void display(Student &s1){
22        cout<<"Inside Diaplay Function !!\n";
23        cout<<"Name of the Student : " <<s1.name <<endl;
24        cout<<"Roll No of the Student : " <<s1.rollNo <<endl;
25        cout<<"Marks of the Student : " <<s1.marks <<endl;
26    }
27 };
28
29 int main(){
30     Student s1;
31     s1.takeInputs(s1);
32     s1.display(s1);
33 }
```

Sample Output

```
1 Inside take inputs Functions !!
2 Enter Name of Student : Navdeep Singh
3 Enter Roll No of Student : 24124073
4 Enter Marks of Student : 95
5 Inside Diaplay Function !!
6 Name of the Student : Navdeep Singh
7 Roll No of the Student : 24124073
8 Marks of the Student : 95
```

1. Write overloaded versions of the function multiply() to: Multiply two integers, Multiply a float and an integer (any order), Multiply two doubles.

Code

```
1 float multiply(int a,int b){
2     return a*b;
3 }
4
5 float multiply(int a,float b){
6     return a*b;
7 }
8
9 float multiply(float a,float b){
10    return a*b;
11 }
12
13 int main1(){
14     cout<<multiply(3.5,3) <<endl;
15     cout<<multiply(3,3) <<endl;
16     cout<<multiply(3.5f,3.5f) <<endl; // By default it is considered as
        double
17 }
```

Sample Output

```
1 9
2 9
3 12.25
```

2. Write overloaded version of the function compare () to, compare two integers, compare two strings, and compare two characters.

Code

```
1 void compare(int a,int b){
2     if(a>b) cout<<a <<" is Greater than " <<b <<endl;
3     else if(a<b) cout<<b<<" is Greater than " <<a <<endl;
4     else cout<<"Both " <<a <<" and " <<b <<" are equal" <<endl;
5 }
6
7 void compare(char a,char b){
8     if(a>b) cout<<a <<" is Greater than " <<b <<endl;
9     else if(a<b) cout<<b<<" is Greater than " <<a <<endl;
10    else cout<<"Both " <<a <<" and " <<b <<" are equal" <<endl;
11 }
12
13 void compare(string a,string b){
14     if(a>b) cout<<a <<" is Greater than " <<b <<endl;
15     else if(a<b) cout<<b<<" is Greater than " <<a <<endl;
16     else cout<<"Both " <<a <<" and " <<b <<" are equal" <<endl;
```

```
17 }
18
19 int main2(){
20     compare('a','b');
21     compare(1,2);
22     compare("Hello","World");
23 }
```

Sample Output

```
1 b is Greater than a
2 2 is Greater than 1
3 World is Greater than Hello
```

3. Write overloaded version of the function reverse () to, reverse an integer, reverse a string, and reverse a float.

Code

```
1 void reverse(int a){
2     int reversedNum = 0;
3     while(a){
4         int rem = a%10;
5         reversedNum = reversedNum * 10 + rem;
6         a /= 10;
7     }
8     cout<<reversedNum<<endl;
9 }
10
11 void reverse(string s){
12     int start = 0;
13     int end = s.size()-1;
14     while(start<=end){
15         swap(s[start],s[end]);
16         start++;
17         end--;
18     }
19     cout<<s<<endl;
20 }
21
22 void reverse(float a){
23     string s = to_string(a);
24     while(!s.empty() && s.back()=='0') s.pop_back();
25     if(!s.empty() && s.back()=='.') s.pop_back();
26     reverse(s);
27 }
28
29
30 int main3(){
31     reverse(201);
32     reverse("Navdeep");
33 }
```

Sample Output

```
1 102
2 peedvaN
```

4. Define a struct `Book` with members `char title [50]`, `char author [50]`, and `float price`. Create three `Book` variables using designated initialization and display their details.

Code

```
1 struct Book{
2     char title[50];
3     char author[50];
4     float price;
5
6     void display(Book &b) {
7         cout << "Book Details:" << endl;
8         cout << "Title: " << b.title << endl;
9         cout << "Author: " << b.author << endl;
10        cout << "Price: " << b.price << endl;
11    }
12 };
13
14 int main4(){
15     Book b1 = {"Metamorphosis","Franz Kafka",106};
16     Book b2 = {"White Nights","Fyodor Dostoevsky",89.5};
17     Book b3 = {"To Kill a Mockingbird","Harper Lee",120.0};
18
19     b1.display(b1);
20     b2.display(b2);
21     b3.display(b3);
22
23     return 0;
24 }
```

Sample Output

```
1 Book Details:
2 Title: Metamorphosis
3 Author: Franz Kafka
4 Price: 106
5 Book Details:
6 Title: White Nights
7 Author: Fyodor Dostoevsky
8 Price: 89.5
9 Book Details:
10 Title: To Kill a Mockingbird
11 Author: Harper Lee
12 Price: 120
```

5. Define a struct named Employee with int empId, char name [50], and float salary. Write a function printEmployee(Employee) to print employee details. Call this function from main() after taking input from the user.

Code

```
1 struct Employee{
2     int empId;
3     char name[50];
4     float salary;
5
6     void printEmployee(Employee emp){
7         cout << "Employee ID: " << emp.empId << endl;
8         cout << "Name: " << emp.name << endl;
9         cout << "Salary: " << emp.salary << endl;
10    }
11 };
12
13
14 int main5(){
15     Employee emp;
16     cout << "Enter Employee ID: ";
17     cin >> emp.empId;
18     cout << "Enter Name: ";
19     cin.ignore();
20     cin.getline(emp.name,50);
21     cout << "Enter Salary: ";
22     cin >> emp.salary;
23     emp.printEmployee(emp);
24     return 0;
25 }
```

Sample Output

```
1 Enter Employee ID: 45
2 Enter Name: Navdeep Singh
3 Enter Salary: 450000
4 Employee ID: 45
5 Name: Navdeep Singh
6 Salary: 450000
```

6. Define a struct Product with int productId, char name [50], float price. Write a program to input details of n products and display the product with the highest price.

Code

```
1 struct Product{
2     int productId;
3     char name[50];
4     float price;
```

```
5 };
6
7 int main6(){
8     int n;
9     cout<<"Enter Number of Products : ";
10    cin>>n;
11    vector<Product>p(n);
12
13    for(int i=0; i<n; i++){
14        cout<<"Enter details for product "<<i+1<<":"<<endl;
15        cout<<"Enter Product ID: ";
16        cin>>p[i].productId;
17        cout<<"Enter Product Name: ";
18        cin>>p[i].name;
19        cout<<"Enter Product Price: ";
20        cin>>p[i].price;
21    }
22
23    Product maxProduct = p[0];
24    for(int i=1; i<n; i++){
25        if(p[i].price > maxProduct.price){
26            maxProduct = p[i];
27        }
28    }
29
30    cout<<"Product with Highest Price: "<<endl;
31    cout<<"ID: "<<maxProduct.productId<<endl;
32    cout<<"Name: "<<maxProduct.name<<endl;
33    cout<<"Price: "<<maxProduct.price<<endl;
34 }
```

Sample Output

```
1 Enter Number of Products : 3
2 Enter details for product 1:
3 Enter Product ID: 34
4 Enter Product Name: ProductA
5 Enter Product Price: 34
6 Enter details for product 2:
7 Enter Product ID: 56
8 Enter Product Name: ProductB
9 Enter Product Price: 50
10 Enter details for product 3:
11 Enter Product ID: 78
12 Enter Product Name: ProductC
13 Enter Product Price: 45
14 Product with Highest Price:
15 ID: 56
16 Name: ProductB
17 Price: 50
```

7. Define two structures Date with day, month, year, and Person with name, Date dob (date of birth). Write a program to input and display a person's name and date of birth.

Code

```
1 struct Date{
2     int day;
3     int month;
4     int year;
5 };
6
7 struct Person{
8     char name[50];
9     Date dob;
10 };
11
12 int main7(){
13     Person p;
14     cout << "Enter Name: ";
15     cin.getline(p.name,50);
16     cout << "Enter Date of Birth (dd mm yyyy): ";
17     cin >> p.dob.day >> p.dob.month >> p.dob.year;
18
19     cout << "Name: " << p.name << endl;
20     cout << "Date of Birth: " << p.dob.day << "/" << p.dob.month << "/"
21         << p.dob.year << endl;
22
23     return 0;
24 }
```

Sample Output

```
1 Enter Name: John Doe
2 Enter Date of Birth (dd mm yyyy): 31 08 2006
3 Name: John Doe
4 Date of Birth: 31/8/2006
```

Calling all main1 , main2 .. etc from main

Code

```
1 int main(){
2     main1();
3     main2();
4     main3();
5     main4();
6     main5();
7     main6();
8     main7();
9 }
```