Assignment No. 02

- **TITLE –** Implement C++ program to demonstrate use of Constructor and Destructor.

- **CODES & OUTPUTS –**

Problem 1: Student Information System

Write a C++ program to create a class Student with data members: roll number, name, and marks.

- Use a parameterized constructor to initialize the student details.
- Display the details using a member function.
- Use a destructor to display a message "Object destroyed" when the object goes out of scope.

```
class Student{
    public:
        int roll_no;
        string name;
        float marks;
        Student(int r, string n, float m){
            roll_no = r;
            name = n;
            marks = m;
        }
        void display(){
            cout<<"Student Information: "<<endl;
            cout<<"Roll No. "<<roll_no<<endl;
            cout<<"Name: "<<name<<endl;
            cout<<"Marks: "<<marks<<endl;
        }
        ~Student(){
            cout<<"Object destroyed"<<endl;
        }
};

int main(){
    int r;
    string n;
    float m;
    cout<<"Enter Roll No.: "<<endl;
    cin>>r;
    cout<<"Enter Name: "<<endl;
    cin>>n;
    cout<<"Enter Marks: "<<endl;
    cin>>m;
    Student s1(r,n,m);
    s1.display();
    return 0;
}
```

Output –

```
Enter Roll No.:
11
Enter Name:
Apurv
```

```
Enter Marks:
93.27
Student Information:
Apurv
Enter Marks:
Apurv
Enter Marks:
93.27
Student Information:
Roll No. 11
Name: Apurv
Marks: 93.27
Object destroyed
```

Problem 2: Bank Account Management

Create a class BankAccount with data members: account number, account holder name, and balance.

- Initialize the details using a constructor.
- Write member functions to deposit and withdraw money.
- When the object is destroyed, the destructor should display a message "Account closed for <account holder>"

```
#include <iostream>
using namespace std;

class BankAccount{
    public:
        int acc_no;
        string acc_holder_name;
        float bal;
        BankAccount(int a, string n, float b){
            acc_no = a;
            acc_holder_name = n;
            bal = b;
        }
        void deposit(float a){
            bal += a;
            cout<<"Amount deposited successfully"<<endl;
            cout<<"Current Balance : "<<bal<<endl;
        }
        void withdraw(float a){
            if(a > bal){
                cout<<"Insufficient Balance"<<endl;
            }
            else{
                bal -= a;
                cout<<"Amount withdrawn successfully"<<endl;
                cout<<"Current Balance : "<<bal<<endl;
            }
        }
        void display(){
            cout<<"Account Details: "<<endl;
            cout<<"Account Number: "<<acc_no<<endl;
            cout<<"Account Holder Name: "<<acc_holder_name<<endl;
```

```
            cout<<"Balance: "<<bal<<endl;
        }
        ~BankAccount(){
            cout<<"Account closed for "<<acc_holder_name<<endl;
        }
};

int main(){
    int a;
    string n;
    float b;
    cout<<"Enter Account Number: "<<endl;
    cin>>a;
    cout<<"Enter Account Holder Name: "<<endl;
    cin>>n;
    cout<<"Enter Initial Balance: "<<endl;
    cin>>b;
    BankAccount ba1(a,n,b);
    ba1.display();
    int ch;
    float amt;
    do{
        cout<<"Choose an option \n 1. Deposit \n 2. Withdraw \n 3. Display
Details \n 4. Exit "<<endl;
        cin>>ch;
        switch(ch){
            case 1:
            cout<<"Enter amount to deposit: "<<endl;
            cin>>amt;
            ba1.deposit(amt);
            break;
            case 2:
            cout<<"Enter amount to withdraw: "<<endl;
            cin>>amt;
            ba1.withdraw(amt);
            break;
            case 3:
            ba1.display();
            break;
            case 4:
            cout<<"Exitting..."<<endl;
            break;
            default:
            cout<<"Invalid Choice"<<endl;
        }
    }while(ch != 4);
    return 0;
}
```

Output –

Enter Account Number:
1001
Enter Account Holder Name:
Apurv

```
Enter Initial Balance:
15000
Account Details:
Account Number: 1001
Account Holder Name: Apurv
Balance: 15000
Choose an option
 1. Deposit
 2. Withdraw
 3. Display Details
 4. Exit
1
Enter amount to deposit:
2000
Amount deposited successfully
Current Balance : 17000
Choose an option
 1. Deposit
 2. Withdraw
 3. Display Details
 4. Exit
2
Enter amount to withdraw:
3200
Amount withdrawn successfully
Current Balance : 13800
Choose an option
 1. Deposit
 2. Withdraw
 3. Display Details
 4. Exit
3
Account Details:
Account Number: 1001
Account Holder Name: Apurv
Balance: 13800
Choose an option
 1. Deposit
 2. Withdraw
 3. Display Details
 4. Exit
4
Exitting...
Account closed for Apurv
```

Problem 3: Employee Payroll

Develop a class Employee with data members: employee ID, name, and salary.

- Use a default constructor and a parameterized constructor for initialization.
- Provide a function to calculate yearly salary.
  Use a destructor to display "Employee record removed" when the object is destroyed.

```
#include<iostream>
using namespace std;
class Employee{
    public:
```

```
        int emp_id;
        string name;
        float salary;
        Employee(){
            default_data();
        }
        void default_data(){
            emp_id = 100;
            name = "John Cena";
            salary = 20000.0;
            display();
            calculate();
        }
        Employee(int id, string n, float s){
            emp_id = id;
            name = n;
            salary = s;
            display();
            calculate();
        }
        void display(){
            cout<<"Employee Details: "<<endl;
            cout<<"Employee ID: "<<emp_id<<endl;
            cout<<"Name: "<<name<<endl;
            cout<<"Salary: "<<salary<<endl;
        }
        void calculate(){
            float annual = salary * 12;
            cout<<"Annual Salary: "<<annual<<endl;
        }
        ~Employee(){
            cout<<"Employee record removed"<<endl;
        }
};

int main(){
    int id, ch;
    string n;
    float sal;
    do{
        cout<<"Choose an option \n 1. Default Employee Constructor \n 2.
Parameterized Employee Constructor \n 3. Exit"<<endl;
        cin>>ch;
        if(ch == 1){
            Employee e1;
        }else if(ch == 2){
            cout<<"Enter employe id: "<<endl;
            cin>>id;
            cout<<"Enter employee name: "<<endl;
            cin>>n;
            cout<<"Enter employee monthly salary: "<<endl;
            cin>>sal;
            Employee e2(id,n,sal);
        }else if(ch == 3){
            cout<<"Exiting..."<<endl;
```

```
        }else{
            cout<<"Invalid choice"<<endl;
        }
    }while(ch != 3);
    return 0;
}
```

Output –

```
Choose an option
 1. Default Employee Constructor
 2. Parameterized Employee Constructor
 3. Exit
1
Employee Details:
Employee ID: 100
Name: John Cena
Salary: 20000
Annual Salary: 240000
Employee record removed
Choose an option
 1. Default Employee Constructor
 2. Parameterized Employee Constructor
 3. Exit
2
Enter employe id:
1001
Enter employee name:
Apurv
Enter employee monthly salary:
15000
Employee Details:
Employee ID: 1001
Name: Apurv
Salary: 15000
Annual Salary: 180000
Employee record removed
Choose an option
 1. Default Employee Constructor
 2. Parameterized Employee Constructor
 3. Exit
3
Exiting...
```

Conclusion – In this assignment, we've implemented programs on constructors and desctructors in C++.