

Lab Questions:

1. Create a class Employee with data members employeeID, name, and salary. Add member functions to read and display these details. Write a program to input and display any number of employees as desired by the user.

```
#include<iostream>
#include<string>
using namespace std;
class Employee{
    int employeeId;
    string name;
    float salary;
public:
    void readData(){
        cout<<"Enter Employee Id:";
        cin>>employeeId;
        cin.ignore();
        cout<<"Enter name:";
        getline(cin,name);
        cout<<"Enter salary:";
        cin>>salary;
    }
    void displayData(){
        cout<<"Employee name:"<<name<<"Employee
Id:"<<employeeId<<"Salary:"<<salary<<endl;
    }
};

int main(){
    int n;
    cout<<"No of Employees you want to enter"<<endl;
    cin>>n;
```

```

Employee* employees= new Employee[n];
for(int i=0;i<n;i++)
{
    cout<<"\nEnter the details of Employee"<<i+1<<":\n";
    employees[i].readData();
}
cout <<"\nEmployee Details:\n";
    for(int i=0;i<n;i++)
    {
        employees[i].displayData();
    }
    delete [] employees;
    return 0;
}

```

Output:

```

Enter the details of Employee1:
Enter Employee Id:001
Enter name:Sachin Jha
Enter salary:23000

Enter the details of Employee2:
Enter Employee Id:002
Enter name:Sarthak Bhattarai
Enter salary:23500

Enter the details of Employee3:
Enter Employee Id:003
Enter name:Mukesh Pandeya
Enter salary:24000

Enter the details of Employee4:
Enter Employee Id:004
Enter name:Nawnit Poudel
Enter salary:24500

Employee Details:
Employee name:Sachin JhaEmployee Id:1Salary:23000
Employee name:Sarthak BhattaraiEmployee Id:2Salary:23500
Employee name:Mukesh PandeyaEmployee Id:3Salary:24000
Employee name:Nawnit PoudelEmployee Id:4Salary:24500

```

2. Design a class Student with members name, roll, and marks. Provide appropriate methods to read and display data.

```
#include <iostream>
#include <string>
using namespace std;
class Student{
string name;
int roll;
float marks;
public:
    void readData(){
        cout<<"Enter Student Name:";
        cin>>name;
        cout<<"Enter Roll No:";
        cin>>roll;
        cout<<"Enter Marks:";
        cin>>marks;
        cin.ignore();
    }
    void displayData(){
        cout<<"Student Name:"<<name<<" Roll No:"<<roll<<" Marks:"<<marks;
    }
}
int main(){
    Student s;
    s.readData();
    cout<<"\nStudent Details:";
    s.displayData();
    return 0;
```

```
}
```

Output:

```
Enter Student Name:Tom  
Enter Roll No:8  
Enter Marks:100  
  
Student Details:Student Name:Tom Roll No:8 Marks:100
```

3. Write a program designing a class Customer with member variables firstName, lastName, address, and phoneNo. Add member functions to accept input and display these variables.

```
#include <iostream>
#include <string>
using namespace std;
class Customer {
string firstName;
string lastName;
string address;
string phoneNo;
public:
void readData() {
cout << "Enter First Name: ";
getline(cin, firstName);
cout << "Enter Last Name: ";
getline(cin, lastName);
cout << "Enter Address: ";
getline(cin, address);
cout << "Enter Phone No: ";
getline(cin, phoneNo);
}
void displayData() {
cout << "\nCustomer Details:\n";
cout << "First Name: " << firstName << endl;
cout << "Last Name: " << lastName << endl;
cout << "Address: " << address << endl;
cout << "Phone No: " << phoneNo << endl;
}
```

```
};  
  
int main() {  
    Customer customer;  
    customer.readData();  
    customer.displayData();  
    return 0;  
}
```

Output:

```
Enter First Name: Shyam  
Enter Last Name: Bahadur  
Enter Address: Humla  
Enter Phone No: 9819673321  
  
Customer Details:  
First Name: Shyam  
Last Name: Bahadur  
Address: Humla  
Phone No: 9819673321
```

3b. Create a class Information to store name and address of students. Store information for two students and write a function to swap the contents of these two objects.

```
#include <iostream>
#include <string>
using namespace std;
class Information {
    string name;
    string address;
public:
    void readData() {
        cout << "Enter name: ";
        getline(cin, name);
        cout << "Enter address: ";
        getline(cin, address);
    }
    void displayData() {
        cout << "Name: " << name << ", Address: " << address << endl;
    }
    void swapInfo(Information &obj) {
        string tempName = name;
        name = obj.name;
        obj.name = tempName;
        string tempAddress = address;
        address = obj.address;
        obj.address = tempAddress;
    }
};

int main() {
    Information s1, s2;
```

```
cout << "Enter details of student 1:\n";
s1.readData();
cout << "\nEnter details of student 2:\n";
s2.readData();
cout << "\nBefore swapping:\n";
cout << "Student 1: ";
s1.displayData();
cout << "Student 2: ";
s2.displayData();
s1.swapInfo(s2);
cout << "\nAfter swapping:\n";
cout << "Student 1: ";
s1.displayData();
cout << "Student 2: ";
s2.displayData();
return 0;
}
```

Output:

```
Enter details of student 1:
Enter name: Sachin Jha
Enter address: Janakpur

Enter details of student 2:
Enter name: Sarthak Bhattarai
Enter address: Biratnagar

Before swapping:
Student 1: Name: Sachin Jha, Address: Janakpur
Student 2: Name: Sarthak Bhattarai, Address: Biratnagar

After swapping:
Student 1: Name: Sarthak Bhattarai, Address: Biratnagar
Student 2: Name: Sachin Jha, Address: Janakpur
```


4. Design a class to represent a bank account with members: depositor's name, account number, and account type. Include member functions to initialize and display the depositor's name and account type.

```
#include <iostream>
#include <string>
using namespace std;
class BankAccount {
    string depositorName;
    long long accountNumber;
    string accountType;
public:
    void initialize() {
        cout << "Enter depositor's name: ";
        getline(cin, depositorName);
        cout << "Enter account number: ";
        cin >> accountNumber;
        cin.ignore();
        cout << "Enter account type: ";
        getline(cin, accountType);
    }
    void display() {
        cout << "\nDepositor's Name: " << depositorName << endl;
        cout << "Account Type: " << accountType << endl;
    }
};

int main() {
    BankAccount account;
    account.initialize();
    account.display();
}
```

```
return 0;  
}
```

Output:

```
Enter depositor's name: Sachin Jha  
Enter account number: 565656787890009  
Enter account type: Saving  
  
Depositor's Name: Sachin Jha  
Account Type: Saving  
  
Process returned 0 (0x0)   execution time : 16.836  
Press any key to continue
```

5. Write a program to design a class Rectangle with constructors and member functions to calculate area and perimeter.

```
#include <iostream>
using namespace std;
class Rectangle {
double length;
double breadth;
public:
Rectangle() {
length = 0;
breadth = 0;
}
Rectangle(double l, double b) {
length = l;
breadth = b;
}
double area() {
return length * breadth;
}
double perimeter() {
return 2 * (length + breadth);
}
void display() {
cout << "Length: " << length << ", Breadth: " << breadth << endl;
cout << "Area: " << area() << endl;
cout << "Perimeter: " << perimeter() << endl;
}
};
int main() {
```

```
double l, w;  
cout << "Enter length and breadth of the rectangle: ";  
cin >> l >> b;  
Rectangle rect(l, b);  
rect.display();  
return(0);  
}
```

Output:

```
Enter length and breadth of the rectangle: 2  
4  
Length: 2, breadth: 4  
Area: 8  
Perimeter: 12  
Length: 1, breadth: 1  
Area: 1  
Perimeter: 4
```

6. Write a program demonstrating that destructors execute in the reverse order of constructors. Display the corresponding object IDs during construction and destruction.

```
#include <iostream>

using namespace std;

class Demons {
    int id;
public:
    Demons(int i) {
        id = i;
        cout << "Constructor called for object " << id << endl;
    }
    ~Demons() {
        cout << "Destructor called for object " << id << endl;
    }
};

int main() {
    cout << "Creating objects...\n";
    Demons d1(1);
    Demons d2(2);
    Demons d3(3);
    cout << "End of main function.\n";
    return 0;
}
```

Output:

```
Creating objects...
Constructor called for object 1
Constructor called for object 2
Constructor called for object 3
End of main function.
Destructor called for object 3
Destructor called for object 2
Destructor called for object 1
```

7. Create a class Time with members for hours, minutes, and seconds. Write a member function AddTime() that adds two Time objects passed as arguments and returns the result.

```
#include <iostream>

using namespace std;

class Time {
    int hours;
    int minutes;
    int seconds;
public:
    void readTime() {
        cout << "Enter hours: ";
        cin >> hours;
        cout << "Enter minutes: ";
        cin >> minutes;
        cout << "Enter seconds: ";
        cin >> seconds;
    }
    void displayTime() {
        cout << hours << "h " << minutes << "m " << seconds << "s" << endl;
    }
    Time AddTime(Time t2) {
        Time temp;
        temp.seconds = seconds + t2.seconds;
        temp.minutes = minutes + t2.minutes + (temp.seconds / 60);
        temp.seconds %= 60;
        temp.hours = hours + t2.hours + (temp.minutes / 60);
        temp.minutes %= 60;
        return temp;
    }
}
```

```

};

int main() {
    Time t1, t2, sum;
    cout << "Enter first time:\n";
    t1.readTime();
    cout << "\nEnter second time:\n";
    t2.readTime();
    sum = t1.AddTime(t2);

    cout << "\nFirst Time: ";
    t1.displayTime();
    cout << "Second Time: ";
    t2.displayTime();
    cout << "Sum of Time: ";
    sum.displayTime();
    return 0;
}

```

Output:

```

Enter first time:
Enter hours: 3
Enter minutes: 56
Enter seconds: 45

Enter second time:
Enter hours: 4
Enter minutes: 44
Enter seconds: 54

First Time: 3h 56m 45s
Second Time: 4h 44m 54s
Sum of Time: 8h 41m 39s

```

8. Design a class LandMeasure that stores Ropani, Ana, Paisa, and Dam. Write a member function to add two LandMeasure objects and return their sum as a new object.

Nepali Land Measurement Reference:

1 Ropani = 16 Ana

1 Ana = 4 Paisa

1 Paisa = 4 Dam

```
#include <iostream>

using namespace std;

class LandMeasure {
    int ropani;
    int ana;
    int paisa;
    int dam;
public:
    void read() {
        cout << "Enter Ropani: ";
        cin >> ropani;
        cout << "Enter Ana: ";
        cin >> ana;
        cout << "Enter Paisa: ";
        cin >> paisa;
        cout << "Enter Dam: ";
        cin >> dam;
    }
    void display() {
        cout << ropani << " Ropani, " << ana << " Ana, " << paisa << " Paisa, " << dam << "Dam"
        << endl;
    }
    LandMeasure add(LandMeasure l2) {
```



```

LandMeasure result;

result.dam = dam + l2.dam;

result.paisa = paisa + l2.paisa + result.dam / 4;

result.dam %= 4;

result.ana = ana + l2.ana + result.paisa / 4;

result.paisa %= 4;

result.ropani = ropani + l2.ropani + result.ana / 16;

result.ana %= 16;

return result;

}

};

int main() {

LandMeasure l1, l2, sum;

cout << "Enter first land measurement:\n";

l1.read();

cout << "\nEnter second land measurement:\n";

l2.read();

sum = l1.add(l2);

cout << "\nFirst Land: ";

l1.display();

cout << "Second Land: ";

l2.display();

cout << "Total Land: ";

sum.display();

return 0;

}

```

```

Enter first land measurement:
Enter Ropani: 4
Enter Ana: 5
Enter Paisa: 6
Enter Dam: 7

Enter second land measurement:
Enter Ropani: 7
Enter Ana: 7
Enter Paisa: 8
Enter Dam: 4

First Land: 4 Ropani, 5 Ana, 6 Paisa, 7Dam
Second Land: 7 Ropani, 7 Ana, 8 Paisa, 4Dam
Total Land: 12 Ropani, 12 Ana, 14 Paisa, 11Dam

```

9. Create a class Employee and add a member function to increase the salary of each employee by 10%. Read and display the details of all employees before and after the salary increase.

```
#include <iostream>

using namespace std;

class Employee {
    int employeeID;

    string name;
    float salary;
public:
    void read() {
        cout << "Enter Employee ID: ";
        cin >> employeeID;
        cout << "Enter Name: ";
        cin.ignore();
        getline(cin, name);
        cout << "Enter Salary: ";
        cin >> salary;
    }
    void display() {
        cout << "ID: " << employeeID << ", Name: " << name << ", Salary: Rs. " << salary << endl;
    }
    void increaseSalary() {
        salary += salary * 0.10;
    }
};

int main() {
    int n;
    cout << "Enter number of employees: ";
```

```

cin >> n;

Employee emp[n];

cout << "\nEnter details of employees:\n";

for(int i = 0; i < n; i++) {

cout << "\nEmployee " << i + 1 << ":\n";

emp[i].read();

}

cout << "\nDetails before salary increase:\n";

for(int i = 0; i < n; i++) {

emp[i].display();

}

for(int i = 0; i < n; i++) {

emp[i].increaseSalary();

}

cout << "\nDetails after 10% salary increase:\n";

for(int i = 0; i < n; i++) {

emp[i].display();

}

return 0;

}

```

Output:

```

Enter details of employees:

Employee 1:
Enter Employee ID: 001
Enter Name: Sachin Jha
Enter Salary: 23000

Employee 2:
Enter Employee ID: 002
Enter Name: Sarthak Bhattarai
Enter Salary: 23500

Employee 3:
Enter Employee ID: 003
Enter Name: Mukesh Pandeya
Enter Salary: 24000

Details before salary increase:
ID: 1, Name: Sachin Jha, Salary: Rs. 23000
ID: 2, Name: Sarthak Bhattarai, Salary: Rs. 23500
ID: 3, Name: Mukesh Pandeya, Salary: Rs. 24000

Details after 10% salary increase:
ID: 1, Name: Sachin Jha, Salary: Rs. 25300
ID: 2, Name: Sarthak Bhattarai, Salary: Rs. 25850
ID: 3, Name: Mukesh Pandeya, Salary: Rs. 26400

```

10. Write a program to find the area of a square and a rectangle using classes.

Initialize the objects dynamically using pointers.

```
#include <iostream>

using namespace std;

class Square {
float side;
public:
void read() {
cout << "Enter the side of the square: ";
cin >> side;
}
float area() {
return side * side;
}
};

class Rectangle {
float length, breadth;
public:
void read() {
cout << "Enter the length and breadth of the rectangle: ";
cin >> length >> breadth;
}
float area() {
return length * breadth;
}
};

int main() {
Square* sq = new Square;
Rectangle* rect = new Rectangle;
cout << "Square:\n";
```

```
sq->read();
cout << "\nRectangle:\n";
rect->read();
cout << "\nArea of Square = " << sq->area() << endl;
cout << "Area of Rectangle = " << rect->area() << endl;
delete sq;
delete rect;
return 0;
}
```

Output:

```
Square:
Enter the side of the square: 3

Rectangle:
Enter the length and breadth of the rectangle: 3
4

Area of Square = 9
Area of Rectangle = 12
```

11. Write a program defining an inline member function `calculateVolume()` outside the class using the `inline` keyword for a class `Box`.

```
#include <iostream>

using namespace std;

class Box {
    float length, breadth, height;
public:
    void read() {
        cout << "Enter length, breadth, and height of the box: ";
        cin >> length >> breadth >> height;
    }
    inline float calculateVolume();
};

inline float Box::calculateVolume() {
    return length * breadth * height;
}

int main() {
    Box b;
    b.read();
    cout << "Volume of the box = " << b.calculateVolume() << endl;
    return 0;
}
```

Output:

```
Enter length, breadth, and height of the box: 3
4
6
Volume of the box = 72
```

DISCUSSIONS

In this lab session, we learned the basics of object-oriented programming in C++ by working with classes, constructors, and destructors. We created classes to define data and functions, then used objects to access them. Constructors were used to initialize objects automatically, and we practiced using default, parameterized, and copy constructors. We also implemented destructors to understand how resources are cleaned up when objects go out of scope. This session gave us a clear understanding of classes, objects, constructors and destructors.

CONCLUSION

This lab made the concepts of classes, constructors, and destructors much clearer. We got hands-on experience with creating and initializing objects, and saw how destructors work when objects go out of scope.