Course code: CSC-284
Lab Report: 06

Submitted To

Tanzina Tasnim Bithi

Department of Computer, Science and Engineering

Submitted By

Name: Md.Pranto Ali
ID: 23303101
Program: BCSE
Section: A
Semester: Fall 2024

Submission date: 22/12/2024

Teacher signature

1. Defining Derivd Class:

main.cpp

Share Run

```cpp
#include <iostream>
using namespace std;
class Animal {
public:
    void display() {
        cout << "This is Animal Class." << endl;
    };
};
class Dog : public Animal {
public:
    void print() {
        cout << "This is Dog Class" << endl;
    };
};
int main() {
    cout << "Md.Pranto Ali" << endl << "ID: 23303101" << endl;
    Dog dogObj;
    dogObj.display();
    dogObj.print();
    return 0;
};

```

Output Clear

```
Md.Pranto Ali
ID: 23303101
This is Animal Class.
This is Dog Class


=== Code Execution Successful ===
```

2. Single Inheritance:

main.cpp

Share | Run

```cpp
#include <iostream>
using namespace std;
class A {
protected:
    string name;
    int id;
public:
    void display(){
        cout << "Name: " << name << endl;
        cout << "ID: " << id << endl;
    };
};
class B : public A {
public:
    B(){
        name = "Md. Pranto Islam";
        id = 23303101;
    }
};
int main() {
    cout << "Name: MD.Pranto Ali" << endl << "ID: 23303101" <<
        endl;
    B b1;
    b1.display();
    return 0;
}
```

Output | Clear

```
Name: MD.Pranto Ali
ID: 23303101
Name: Md. Pranto Islam
ID: 23303101


=== Code Execution Successful ===
```

3. Multilevel Inheritance:

main.cpp | Share | Run

```cpp
#include <iostream>
using namespace std;
class A {
public:
    int a;
    void get_A_data(){
        cout << "Enter value of a: ";
        cin >> a;
    };
};
class B : public A {
public:
    int b;
    void get_B_data(){
        cout << "Enter value of b: ";
        cin >> b;
    };
};
class C : public B {
private:
    int c;
public:
    void get_C_data(){
        cout << "Enter value of c: ";
        cin >> c;
    };
    // function to print sum
    void sum(){
        int ans = a + b + c;
        cout << "sum: " << ans;
```

Output | Clear

```
Md.Pranto Ali
ID: 23303101
Enter value of a: 5
Enter value of b: 6
Enter value of c: 7
sum: 18

=== Code Execution Successful ===
```

4. Multiple Inheritance:

main.cpp Share Run Output Clear

```cpp
#include <iostream>
using namespace std;
class A {
protected:
    int a;
public:
    void get_a(int n){
        a = n;
    };
};
class B {
protected:
    int b;
public:
    void get_b(int n){
        b = n;
    };
};
class C : public A, public B {
public:
    void display() {
        cout << "The value of a is : " << a << endl;  // First Bass Class
        cout << "The value of b is : " << b << endl;  // Second Bass Class
        cout << "Product of a and b is : " << a * b;  // Multiplication of a and b and Display
    };
};
int main() {
```

```
Name: MD.Pranto Ali
ID: 23303101
The value of a is : 10
The value of b is : 20
Product of a and b is : 200

=== Code Execution Successful ===
```

5. Hierarchical Inheritance:

main.cpp

Share Run

```
class A{
public:
    void show_A() {
        cout<<"class A"<<endl;
    };
};
class B : public A{
public:
    void show_B() {
        cout<<"class B"<<endl;
    };
};
class C : public A{
public:
    void show_C() {
        cout<<"class C"<<endl;
    };
};
int main() {
    cout << "Md.Pranto Ali" << endl << "ID: 23303101" << endl;
    B b;
    cout<<"calling from B: "<<endl;
    b.show_B();
    b.show_A();

    C c;
    cout<<"calling from C: "<<endl;
    c.show_C();
    c.show_A();
return 0;
};
```

Output Clear

```
Md.Pranto Ali
ID: 23303101
calling from B:
class B
class A
calling from C:
class C
class A


=== Code Execution Successful ===
```

6. Hybrid Inheritance

main.cpp

Share Run

```cpp
#include <iostream>
using namespace std;
class Person {
protected:
    string name;
public:
    Person(string name): name(name){};
    void display(){
        cout << "Name: " << name << endl;
    };
};
class Employee : public Person {
protected:
    int employeeId;
public:
    Employee(string name, int id): Person(name), employeeId(id
        ){};
    void displayEmployee(){
        display();
        cout << "Employee ID: " << employeeId << endl;
        cout << "Method inside Derived Class Employee"<< endl;
    };
};
class Student : public Person {
protected:
    int studentId;

public:
    Student(string name, int id) : Person(name), studentId(id){}
        ;
```

Output Clear

```
Md.Pranto Ali
ID: 23303101
Methods inside Derived Class StudentIntern :
Name: Riya
Employee ID: 67537
Method inside Derived Class Employee
Name: Riya
Student ID: 2215
Method inside Derived Class Student


=== Code Execution Successful ===
```