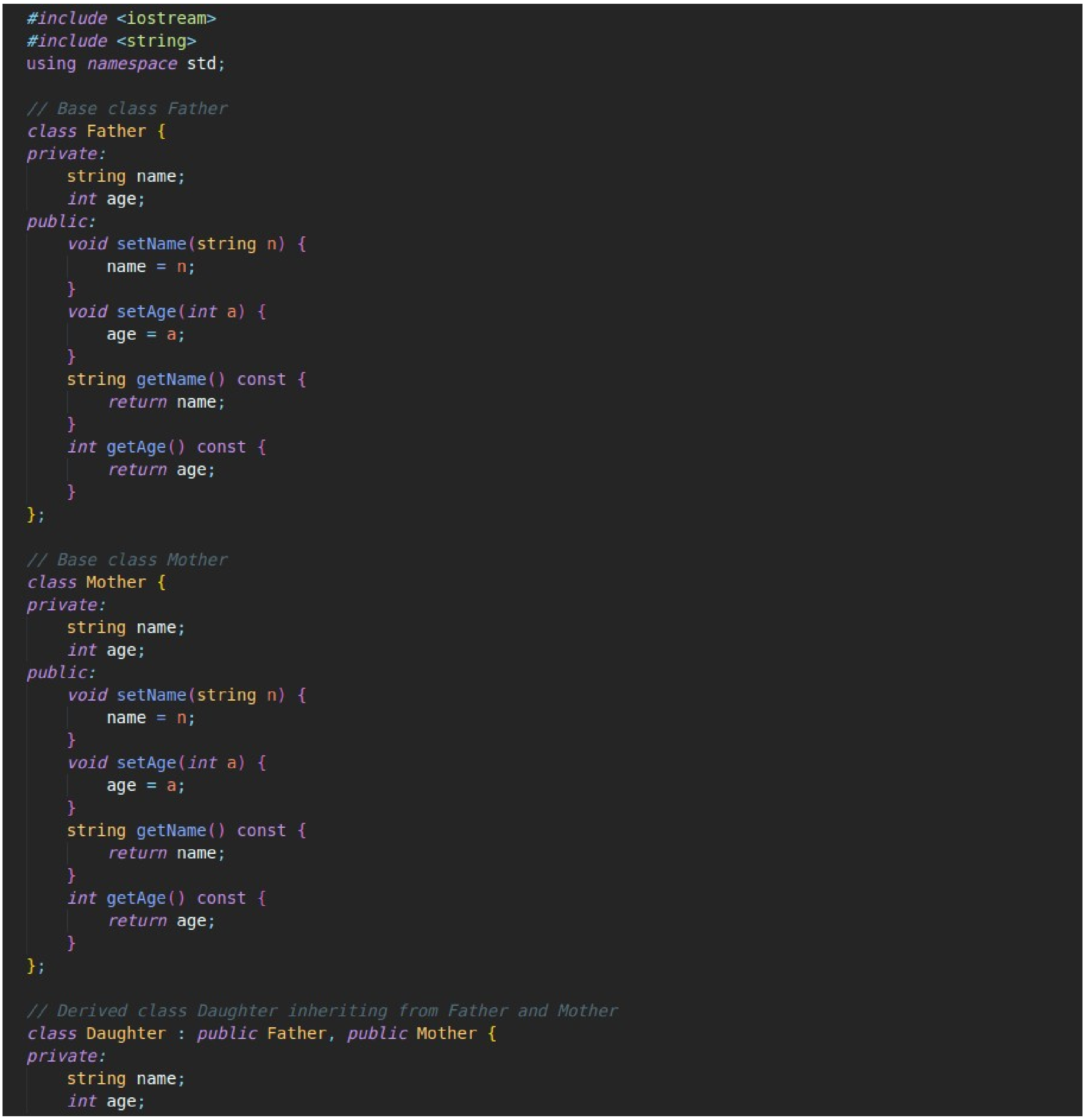
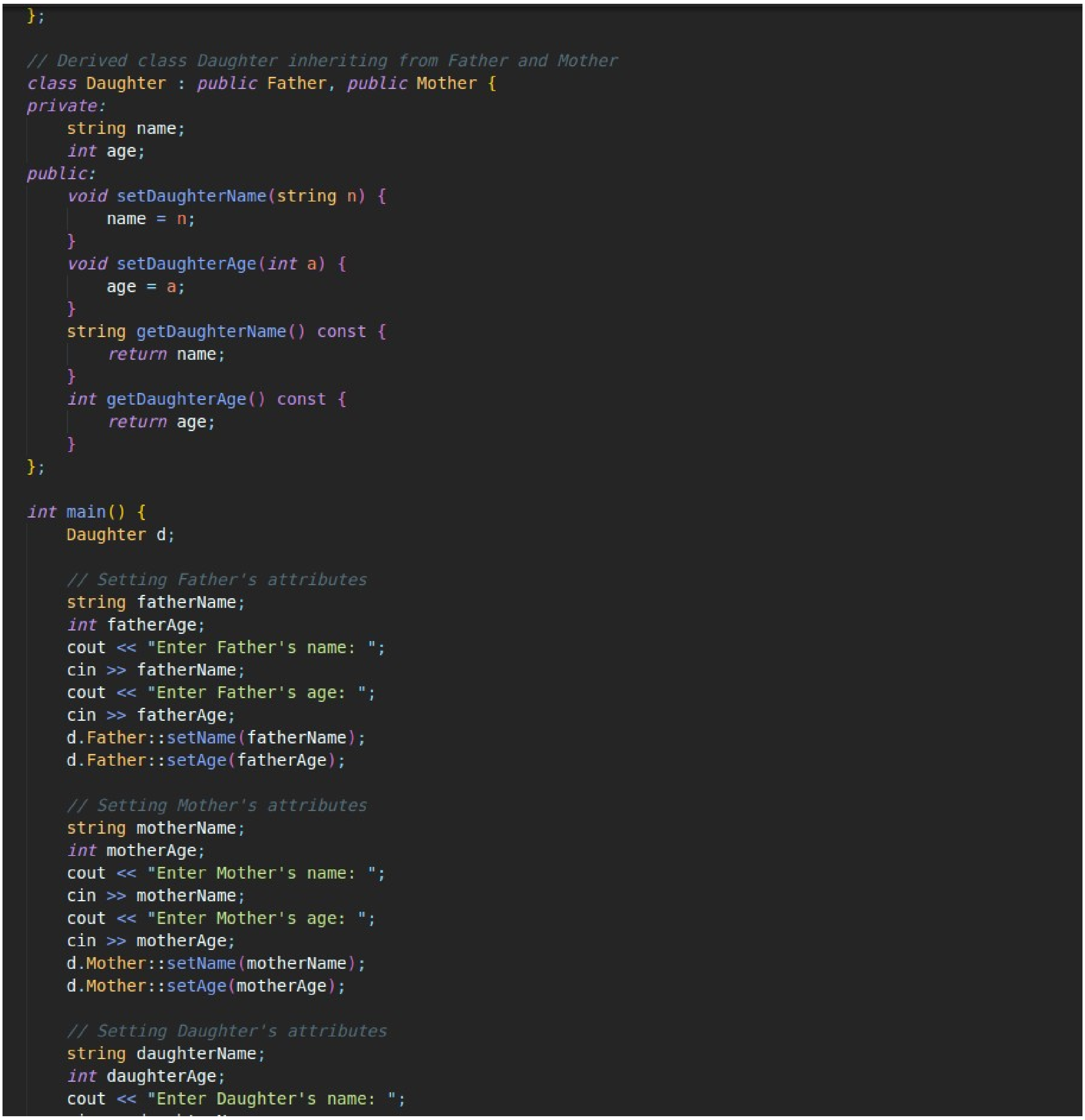
Nithin S
221IT085

IT150 End Sem Lab Assignment

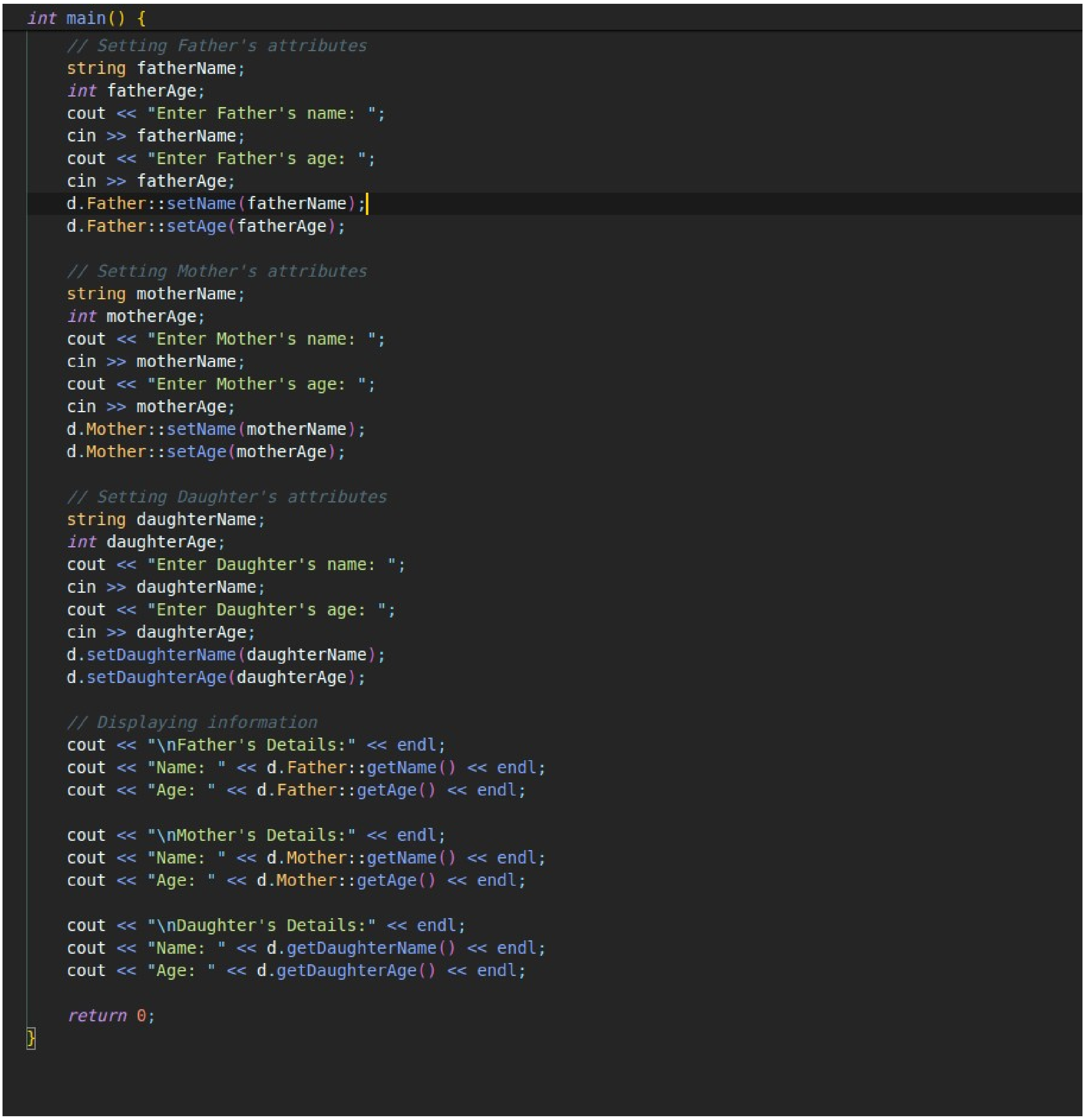
Q1.



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

// Base class Father
class Father {
private:
    string name;
    int age;
public:
    void setName(string n) {
        name = n;
    }
    void setAge(int a) {
        age = a;
    }
    string getName() const {
        return name;
    }
    int getAge() const {
        return age;
    }
};

// Base class Mother
class Mother {
private:
    string name;
    int age;
public:
    void setName(string n) {
        name = n;
    }
    void setAge(int a) {
        age = a;
    }
    string getName() const {
        return name;
    }
    int getAge() const {
        return age;
    }
};

// Derived class Daughter inheriting from Father and Mother
class Daughter : public Father, public Mother {
private:
    string name;
    int age;
```

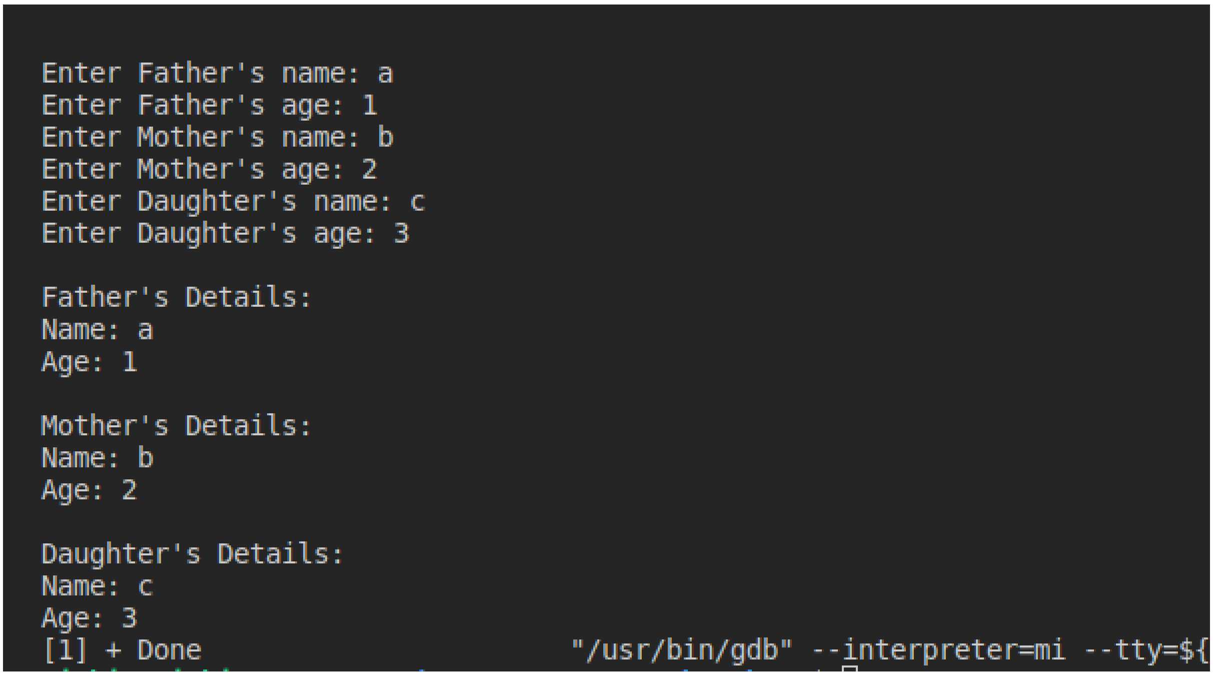


```cpp
};

// Derived class Daughter inheriting from Father and Mother
class Daughter : public Father, public Mother {
private:
    string name;
    int age;
public:
    void setDaughterName(string n) {
        name = n;
    }
    void setDaughterAge(int a) {
        age = a;
    }
    string getDaughterName() const {
        return name;
    }
    int getDaughterAge() const {
        return age;
    }
};

int main() {
    Daughter d;

    // Setting Father's attributes
    string fatherName;
    int fatherAge;
    cout << "Enter Father's name: ";
    cin >> fatherName;
    cout << "Enter Father's age: ";
    cin >> fatherAge;
    d.Father::setName(fatherName);
    d.Father::setAge(fatherAge);

    // Setting Mother's attributes
    string motherName;
    int motherAge;
    cout << "Enter Mother's name: ";
    cin >> motherName;
    cout << "Enter Mother's age: ";
    cin >> motherAge;
    d.Mother::setName(motherName);
    d.Mother::setAge(motherAge);

    // Setting Daughter's attributes
    string daughterName;
    int daughterAge;
    cout << "Enter Daughter's name: ";
```



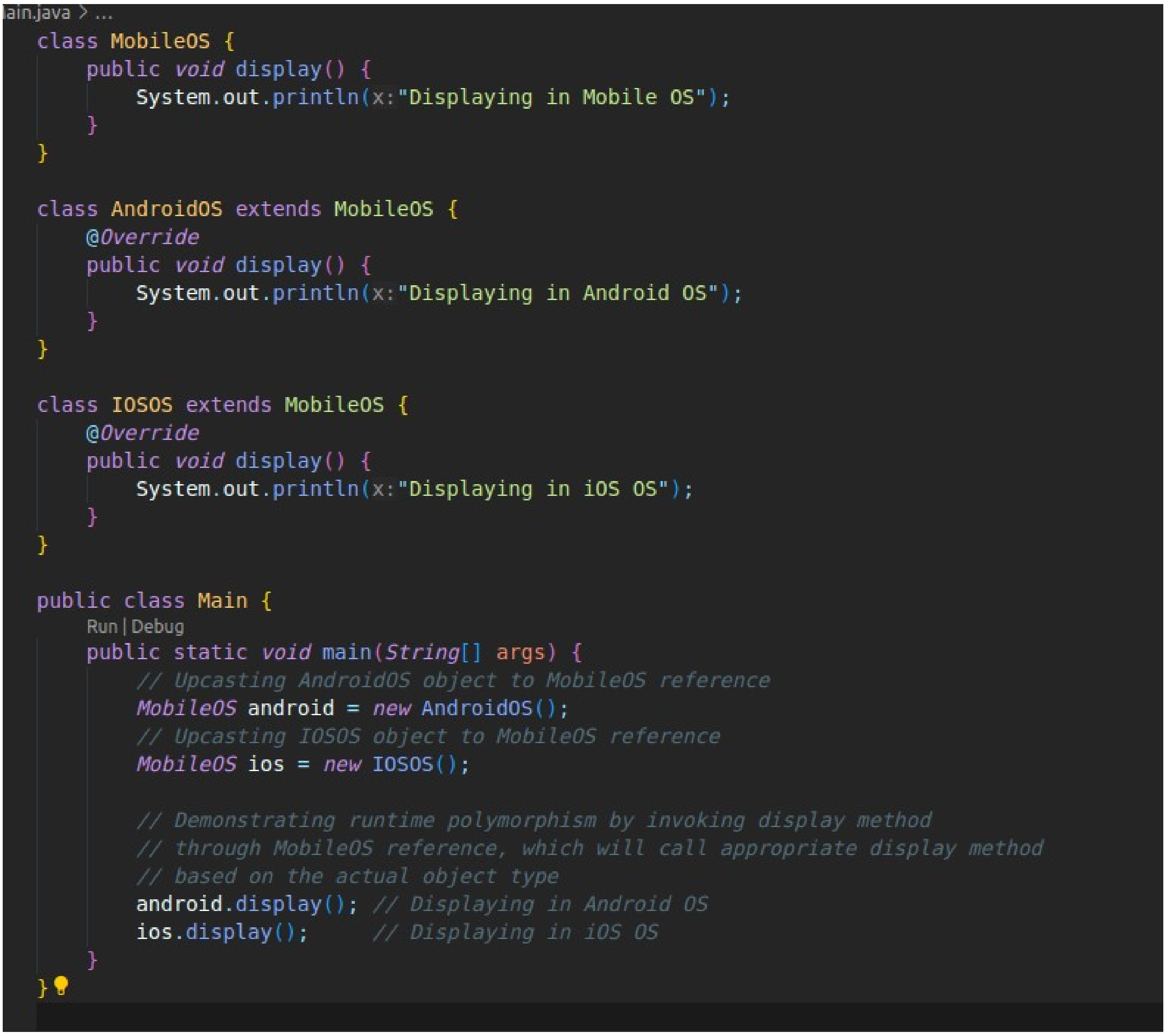
```cpp
int main() {
    // Setting Father's attributes
    string fatherName;
    int fatherAge;
    cout << "Enter Father's name: ";
    cin >> fatherName;
    cout << "Enter Father's age: ";
    cin >> fatherAge;
    d.Father::setName(fatherName);
    d.Father::setAge(fatherAge);

    // Setting Mother's attributes
    string motherName;
    int motherAge;
    cout << "Enter Mother's name: ";
    cin >> motherName;
    cout << "Enter Mother's age: ";
    cin >> motherAge;
    d.Mother::setName(motherName);
    d.Mother::setAge(motherAge);

    // Setting Daughter's attributes
    string daughterName;
    int daughterAge;
    cout << "Enter Daughter's name: ";
    cin >> daughterName;
    cout << "Enter Daughter's age: ";
    cin >> daughterAge;
    d.setDaughterName(daughterName);
    d.setDaughterAge(daughterAge);

    // Displaying information
    cout << "\nFather's Details:" << endl;
    cout << "Name: " << d.Father::getName() << endl;
    cout << "Age: " << d.Father::getAge() << endl;

    cout << "\nMother's Details:" << endl;
    cout << "Name: " << d.Mother::getName() << endl;
    cout << "Age: " << d.Mother::getAge() << endl;

    cout << "\nDaughter's Details:" << endl;
    cout << "Name: " << d.getDaughterName() << endl;
    cout << "Age: " << d.getDaughterAge() << endl;

    return 0;
}
```

OUTPUT



```
Enter Father's name: a
Enter Father's age: 1
Enter Mother's name: b
Enter Mother's age: 2
Enter Daughter's name: c
Enter Daughter's age: 3

Father's Details:
Name: a
Age: 1

Mother's Details:
Name: b
Age: 2

Daughter's Details:
Name: c
Age: 3
[1] + Done                       "/usr/bin/gdb" --interpreter=mi --tty=${
```

Q2.



```java
class MobileOS {
    public void display() {
        System.out.println(x:"Displaying in Mobile OS");
    }
}

class AndroidOS extends MobileOS {
    @Override
    public void display() {
        System.out.println(x:"Displaying in Android OS");
    }
}

class IOSOS extends MobileOS {
    @Override
    public void display() {
        System.out.println(x:"Displaying in iOS OS");
    }
}

public class Main {
    public static void main(String[] args) {
        // Upcasting AndroidOS object to MobileOS reference
        MobileOS android = new AndroidOS();
        // Upcasting IOSOS object to MobileOS reference
        MobileOS ios = new IOSOS();

        // Demonstrating runtime polymorphism by invoking display method
        // through MobileOS reference, which will call appropriate display method
        // based on the actual object type
        android.display(); // Displaying in Android OS
        ios.display();     // Displaying in iOS OS
    }
}
```