OBJECT ORIENTED PROGRAMMING

[CT-260]

NAME: TAHA AHMED MALLICK

ROLL NO: CT-25183

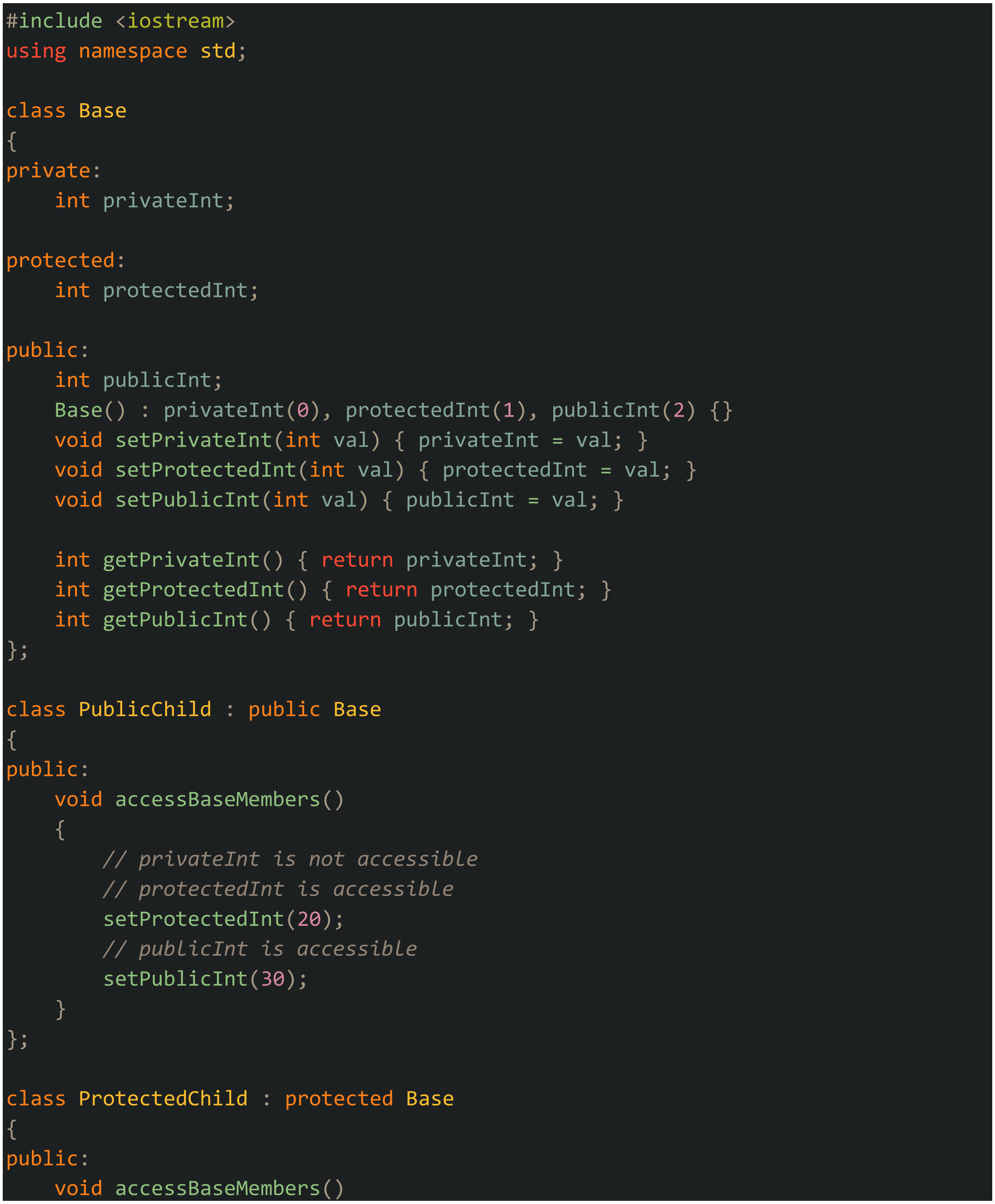
DEPARTMENT: BCIT **BATCH:** 2025

YEAR & SECTION: FSCS-D

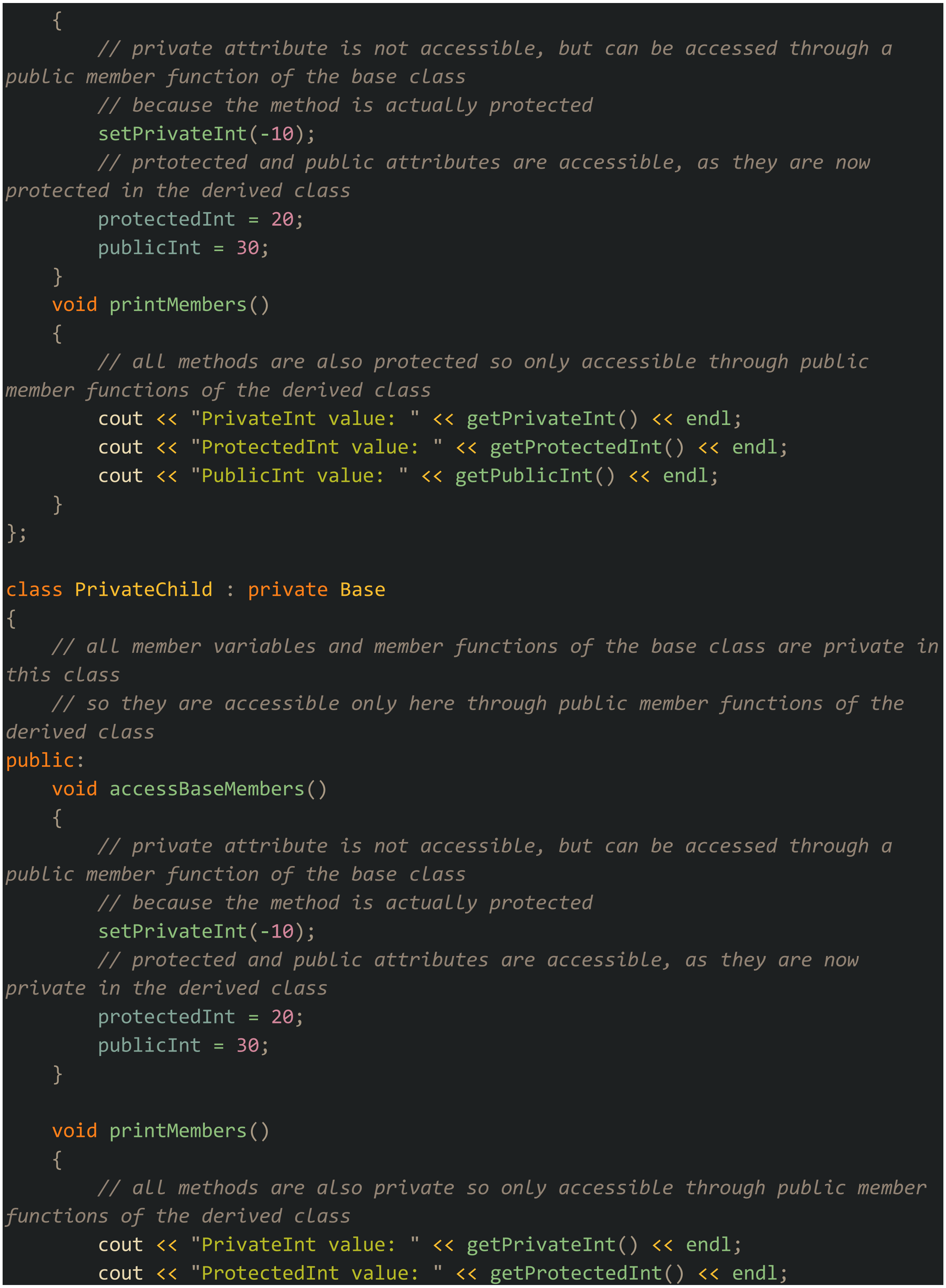
LAB :5

QUESTION#1:

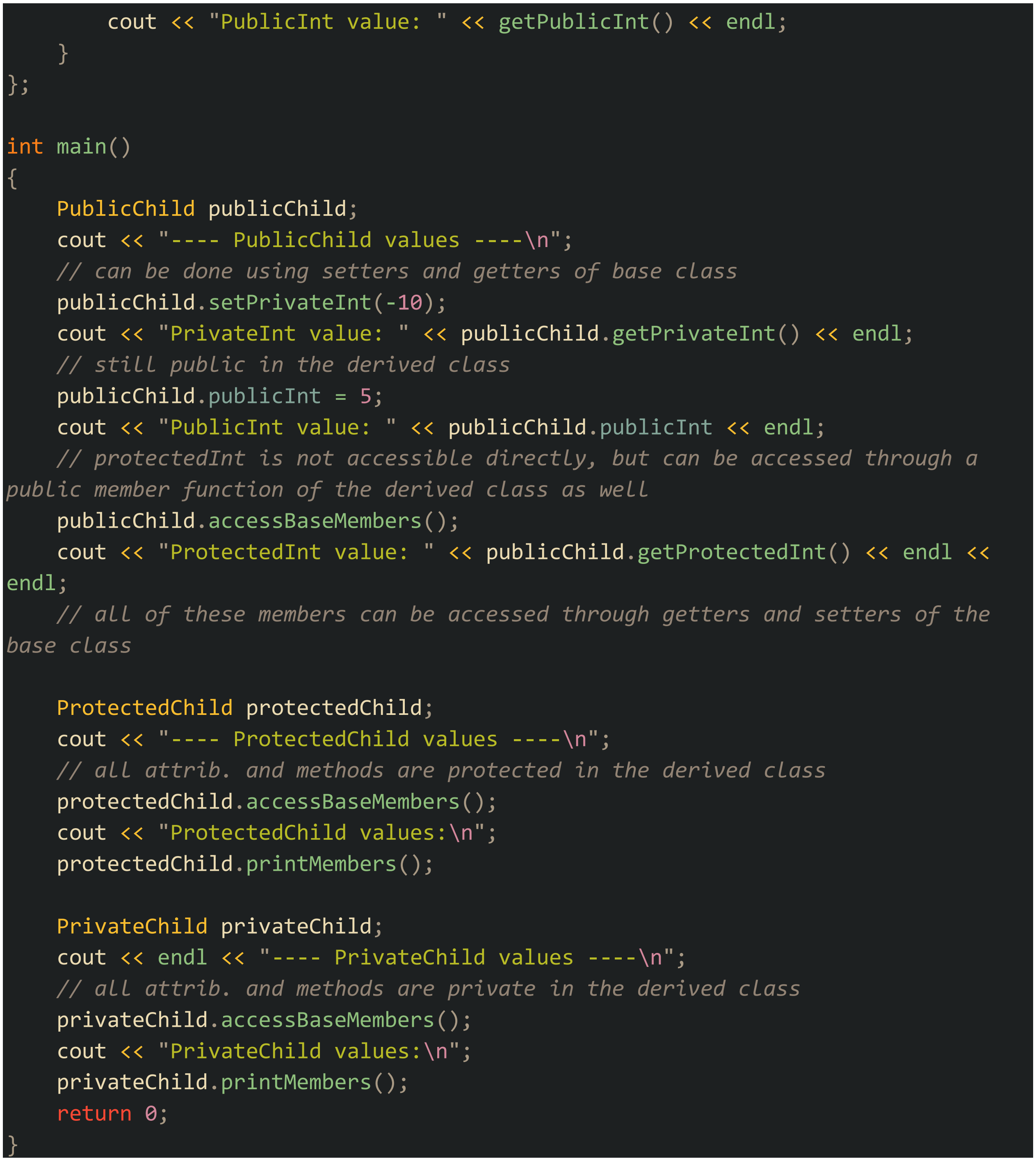
CODE:



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

class Base
{
private:
    int privateInt;

protected:
    int protectedInt;

public:
    int publicInt;
    Base() : privateInt(0), protectedInt(1), publicInt(2) {}
    void setPrivateInt(int val) { privateInt = val; }
    void setProtectedInt(int val) { protectedInt = val; }
    void setPublicInt(int val) { publicInt = val; }

    int getPrivateInt() { return privateInt; }
    int getProtectedInt() { return protectedInt; }
    int getPublicInt() { return publicInt; }
};

class PublicChild : public Base
{
public:
    void accessBaseMembers()
    {
        // privateInt is not accessible
        // protectedInt is accessible
        setProtectedInt(20);
        // publicInt is accessible
        setPublicInt(30);
    }
};

class ProtectedChild : protected Base
{
public:
    void accessBaseMembers()
```

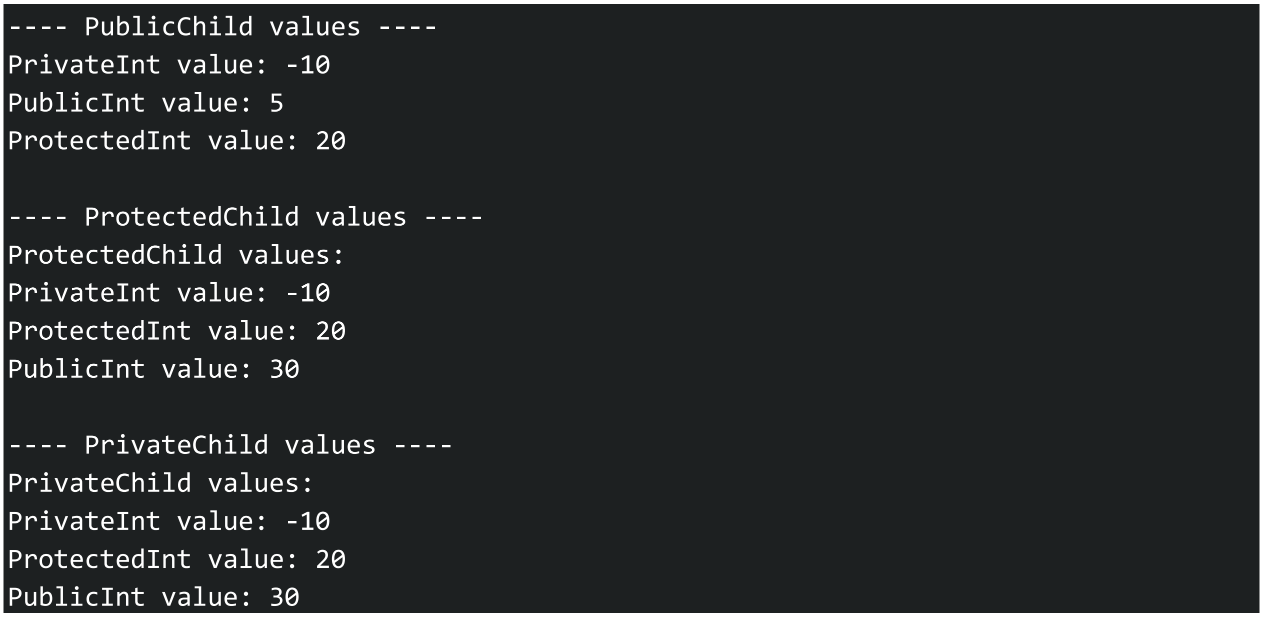


```
    {
        // private attribute is not accessible, but can be accessed through a
public member function of the base class
        // because the method is actually protected
        setPrivateInt(-10);
        // prtotected and public attributes are accessible, as they are now
protected in the derived class
        protectedInt = 20;
        publicInt = 30;
    }
    void printMembers()
    {
        // all methods are also protected so only accessible through public
member functions of the derived class
        cout << "PrivateInt value: " << getPrivateInt() << endl;
        cout << "ProtectedInt value: " << getProtectedInt() << endl;
        cout << "PublicInt value: " << getPublicInt() << endl;
    }
};

class PrivateChild : private Base
{
    // all member variables and member functions of the base class are private in
this class
    // so they are accessible only here through public member functions of the
derived class
public:
    void accessBaseMembers()
    {
        // private attribute is not accessible, but can be accessed through a
public member function of the base class
        // because the method is actually protected
        setPrivateInt(-10);
        // protected and public attributes are accessible, as they are now
private in the derived class
        protectedInt = 20;
        publicInt = 30;
    }

    void printMembers()
    {
        // all methods are also private so only accessible through public member
functions of the derived class
        cout << "PrivateInt value: " << getPrivateInt() << endl;
        cout << "ProtectedInt value: " << getProtectedInt() << endl;
```



```cpp
        cout << "PublicInt value: " << getPublicInt() << endl;
    }
};

int main()
{
    PublicChild publicChild;
    cout << "---- PublicChild values ----\n";
    // can be done using setters and getters of base class
    publicChild.setPrivateInt(-10);
    cout << "PrivateInt value: " << publicChild.getPrivateInt() << endl;
    // still public in the derived class
    publicChild.publicInt = 5;
    cout << "PublicInt value: " << publicChild.publicInt << endl;
    // protectedInt is not accessible directly, but can be accessed through a public member function of the derived class as well
    publicChild.accessBaseMembers();
    cout << "ProtectedInt value: " << publicChild.getProtectedInt() << endl << endl;
    // all of these members can be accessed through getters and setters of the base class

    ProtectedChild protectedChild;
    cout << "---- ProtectedChild values ----\n";
    // all attrib. and methods are protected in the derived class
    protectedChild.accessBaseMembers();
    cout << "ProtectedChild values:\n";
    protectedChild.printMembers();

    PrivateChild privateChild;
    cout << endl << "---- PrivateChild values ----\n";
    // all attrib. and methods are private in the derived class
    privateChild.accessBaseMembers();
    cout << "PrivateChild values:\n";
    privateChild.printMembers();
    return 0;
}
```

OUTPUT:

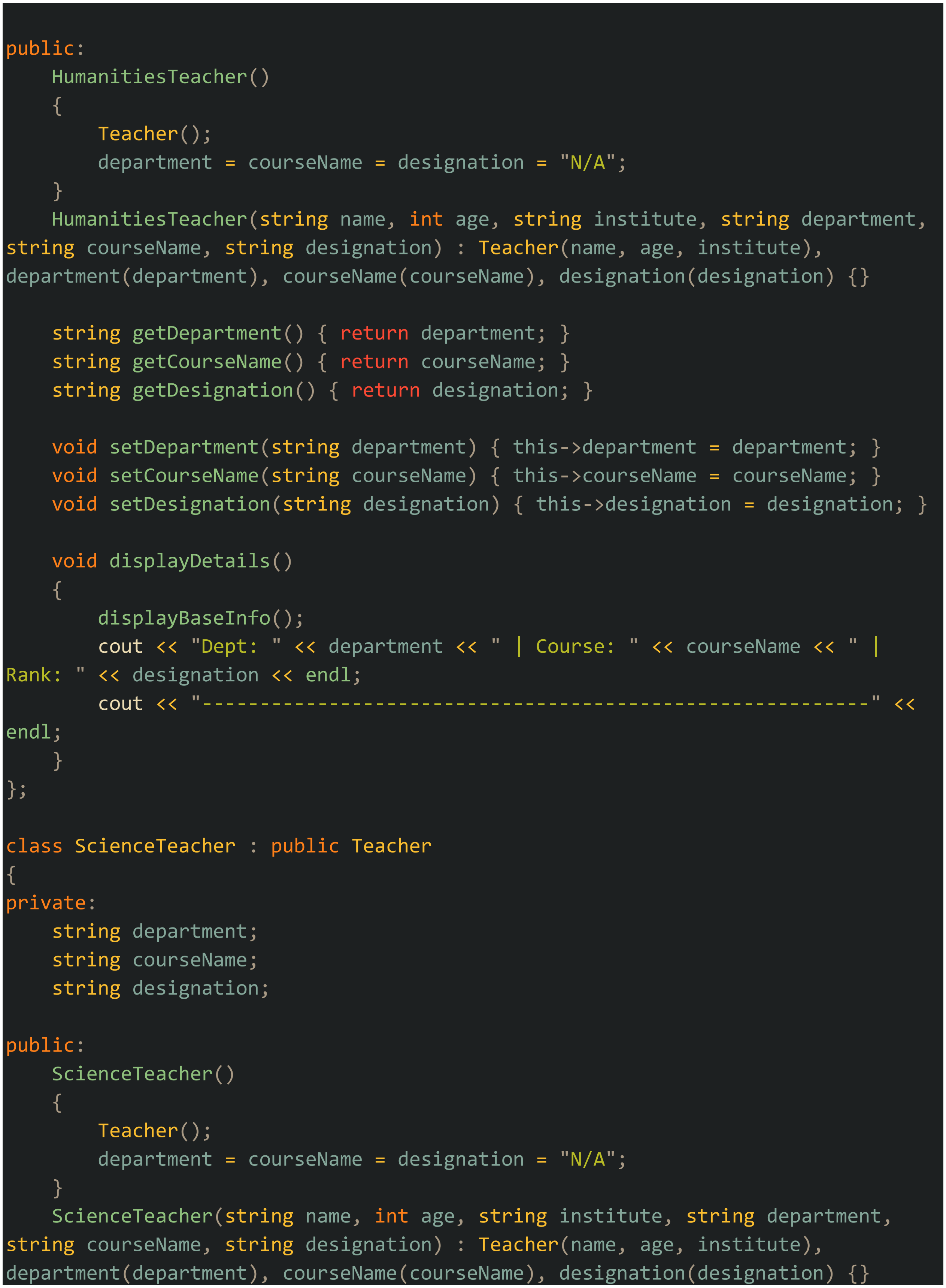


```
---- PublicChild values ----
PrivateInt value: -10
PublicInt value: 5
ProtectedInt value: 20

---- ProtectedChild values ----
ProtectedChild values:
PrivateInt value: -10
ProtectedInt value: 20
PublicInt value: 30

---- PrivateChild values ----
PrivateChild values:
PrivateInt value: -10
ProtectedInt value: 20
PublicInt value: 30
```

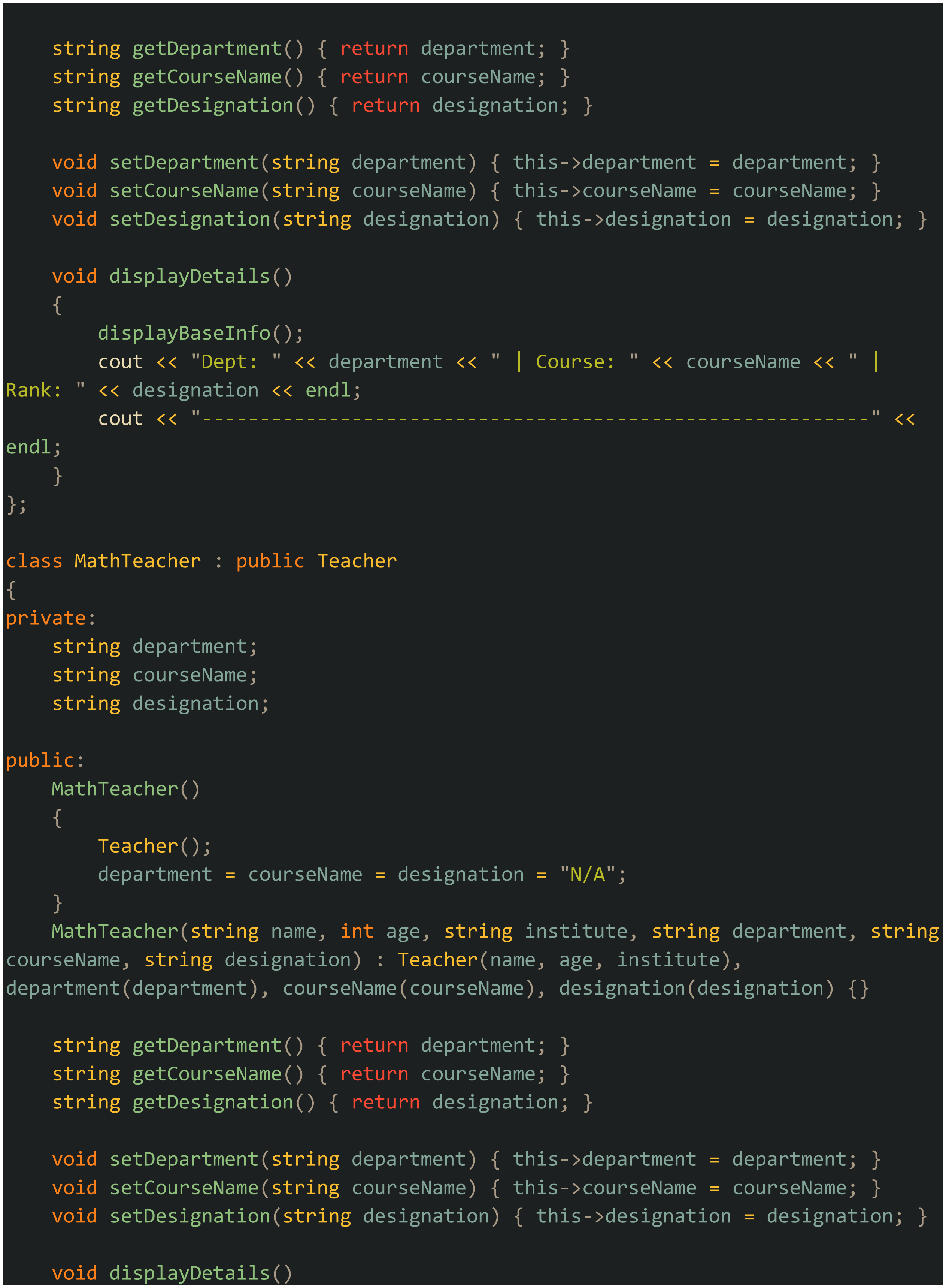
QUESTION#2:

CODE:

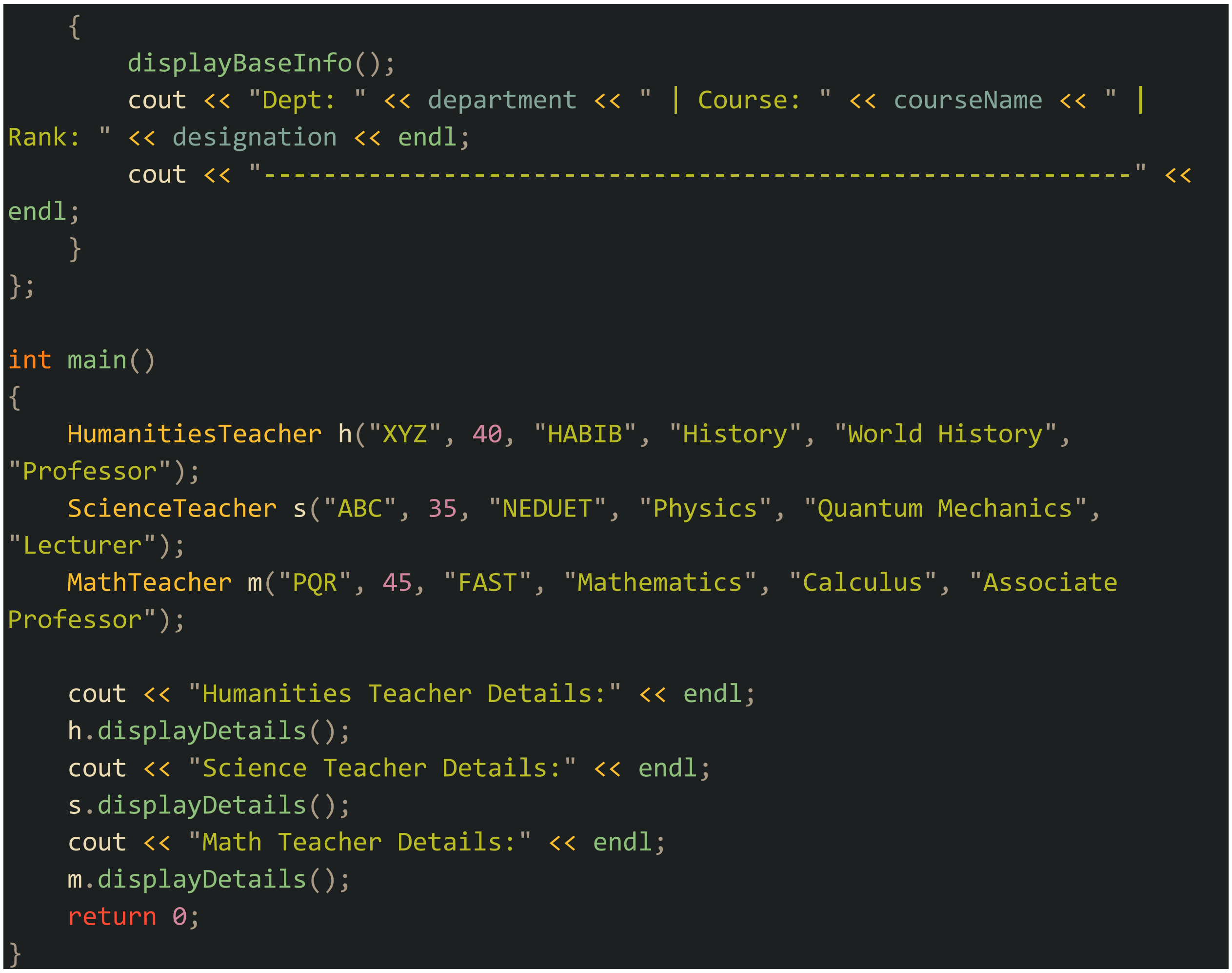
```
#include <iostream>
using namespace std;

class Teacher
{
private:
    string name;
    int age;
    string institute;

public:
    Teacher()
    {
        name = institute = "N/A";
        age = 0;
    }
    Teacher(string name, int age, string institute) : name(name), age(age),
institute(institute) {}

    int getAge() { return age; }
    string getName() { return name; }
    string getInstitute() { return institute; }

    void setAge(int age) { this->age = age; }
    void setName(string name) { this->name = name; }
    void setInstitute(string institute) { this->institute = institute; }

    void displayBaseInfo()
    {
        cout << "Name: " << name << " | Age: " << age << " | Institute: " <<
institute << endl;
    }
};

class HumanitiesTeacher : public Teacher
{
private:
    string department;
    string courseName;
    string designation;
```



```cpp
public:
    HumanitiesTeacher()
    {
        Teacher();
        department = courseName = designation = "N/A";
    }
    HumanitiesTeacher(string name, int age, string institute, string department, string courseName, string designation) : Teacher(name, age, institute), department(department), courseName(courseName), designation(designation) {}

    string getDepartment() { return department; }
    string getCourseName() { return courseName; }
    string getDesignation() { return designation; }

    void setDepartment(string department) { this->department = department; }
    void setCourseName(string courseName) { this->courseName = courseName; }
    void setDesignation(string designation) { this->designation = designation; }

    void displayDetails()
    {
        displayBaseInfo();
        cout << "Dept: " << department << " | Course: " << courseName << " | Rank: " << designation << endl;
        cout << "-----------------------------------------------------------" << endl;
    }
};

class ScienceTeacher : public Teacher
{
private:
    string department;
    string courseName;
    string designation;

public:
    ScienceTeacher()
    {
        Teacher();
        department = courseName = designation = "N/A";
    }
    ScienceTeacher(string name, int age, string institute, string department, string courseName, string designation) : Teacher(name, age, institute), department(department), courseName(courseName), designation(designation) {}
```

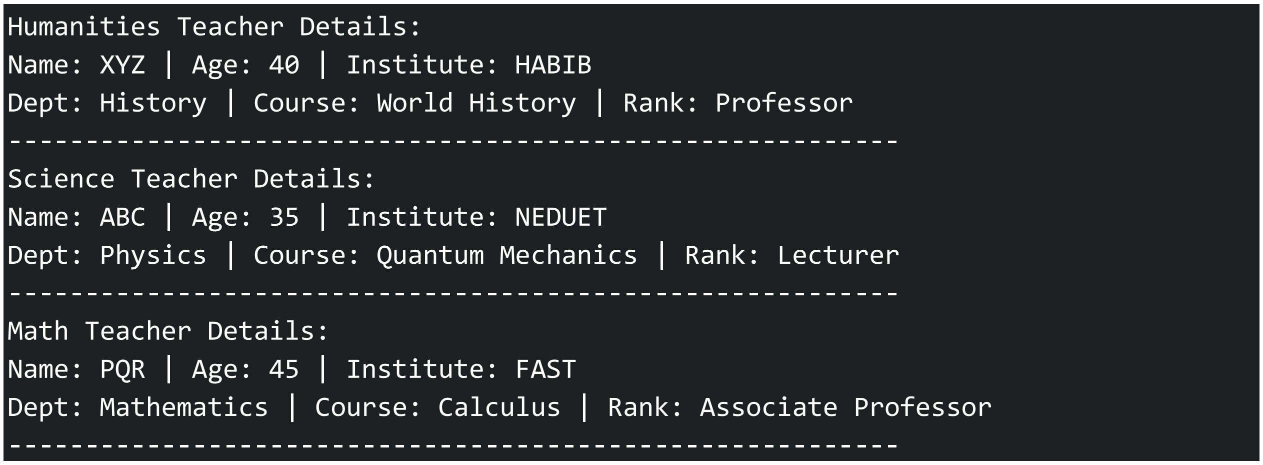


```
    string getDepartment() { return department; }
    string getCourseName() { return courseName; }
    string getDesignation() { return designation; }

    void setDepartment(string department) { this->department = department; }
    void setCourseName(string courseName) { this->courseName = courseName; }
    void setDesignation(string designation) { this->designation = designation; }

    void displayDetails()
    {
        displayBaseInfo();
        cout << "Dept: " << department << " | Course: " << courseName << " |
Rank: " << designation << endl;
        cout << "----------------------------------------------------------" <<
endl;
    }
};

class MathTeacher : public Teacher
{
private:
    string department;
    string courseName;
    string designation;

public:
    MathTeacher()
    {
        Teacher();
        department = courseName = designation = "N/A";
    }
    MathTeacher(string name, int age, string institute, string department, string
courseName, string designation) : Teacher(name, age, institute),
department(department), courseName(courseName), designation(designation) {}

    string getDepartment() { return department; }
    string getCourseName() { return courseName; }
    string getDesignation() { return designation; }

    void setDepartment(string department) { this->department = department; }
    void setCourseName(string courseName) { this->courseName = courseName; }
    void setDesignation(string designation) { this->designation = designation; }

    void displayDetails()
```



```
    {
        displayBaseInfo();
        cout << "Dept: " << department << " | Course: " << courseName << " |
Rank: " << designation << endl;
        cout << "----------------------------------------------------------" <<
endl;
    }
};

int main()
{
    HumanitiesTeacher h("XYZ", 40, "HABIB", "History", "World History",
"Professor");
    ScienceTeacher s("ABC", 35, "NEDUET", "Physics", "Quantum Mechanics",
"Lecturer");
    MathTeacher m("PQR", 45, "FAST", "Mathematics", "Calculus", "Associate
Professor");

    cout << "Humanities Teacher Details:" << endl;
    h.displayDetails();
    cout << "Science Teacher Details:" << endl;
    s.displayDetails();
    cout << "Math Teacher Details:" << endl;
    m.displayDetails();
    return 0;
}
```

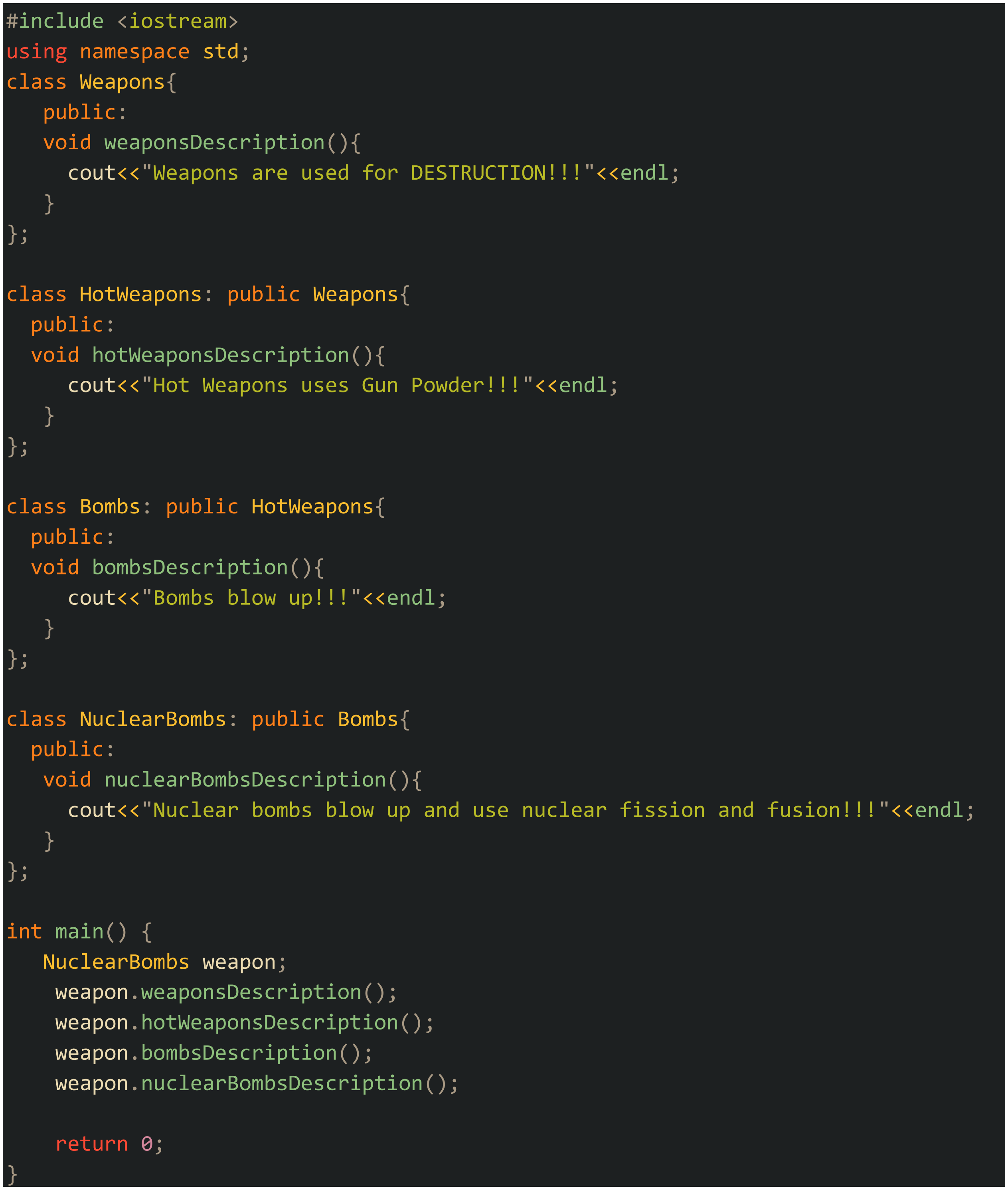
OUTPUT:



```
Humanities Teacher Details:
Name: XYZ | Age: 40 | Institute: HABIB
Dept: History | Course: World History | Rank: Professor
----------------------------------------------------------
Science Teacher Details:
Name: ABC | Age: 35 | Institute: NEDUET
Dept: Physics | Course: Quantum Mechanics | Rank: Lecturer
----------------------------------------------------------
Math Teacher Details:
Name: PQR | Age: 45 | Institute: FAST
Dept: Mathematics | Course: Calculus | Rank: Associate Professor
----------------------------------------------------------
```

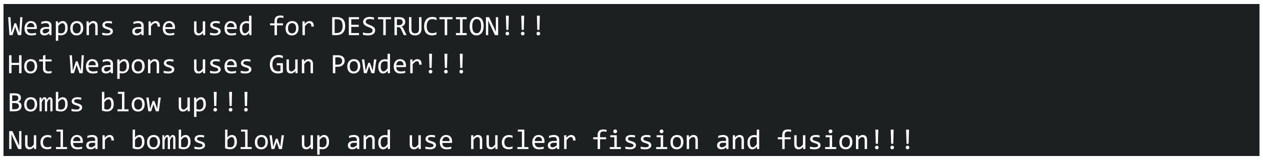
QUESTION#3:

CODE:



```cpp
#include <iostream>
using namespace std;
class Weapons{
   public:
   void weaponsDescription(){
     cout<<"Weapons are used for DESTRUCTION!!!"<<endl;
   }
};

class HotWeapons: public Weapons{
  public:
  void hotWeaponsDescription(){
     cout<<"Hot Weapons uses Gun Powder!!!"<<endl;
   }
};

class Bombs: public HotWeapons{
  public:
  void bombsDescription(){
     cout<<"Bombs blow up!!!"<<endl;
   }
};

class NuclearBombs: public Bombs{
  public:
   void nuclearBombsDescription(){
     cout<<"Nuclear bombs blow up and use nuclear fission and fusion!!!"<<endl;
   }
};

int main() {
   NuclearBombs weapon;
    weapon.weaponsDescription();
    weapon.hotWeaponsDescription();
    weapon.bombsDescription();
    weapon.nuclearBombsDescription();

    return 0;
}
```

OUTPUT:

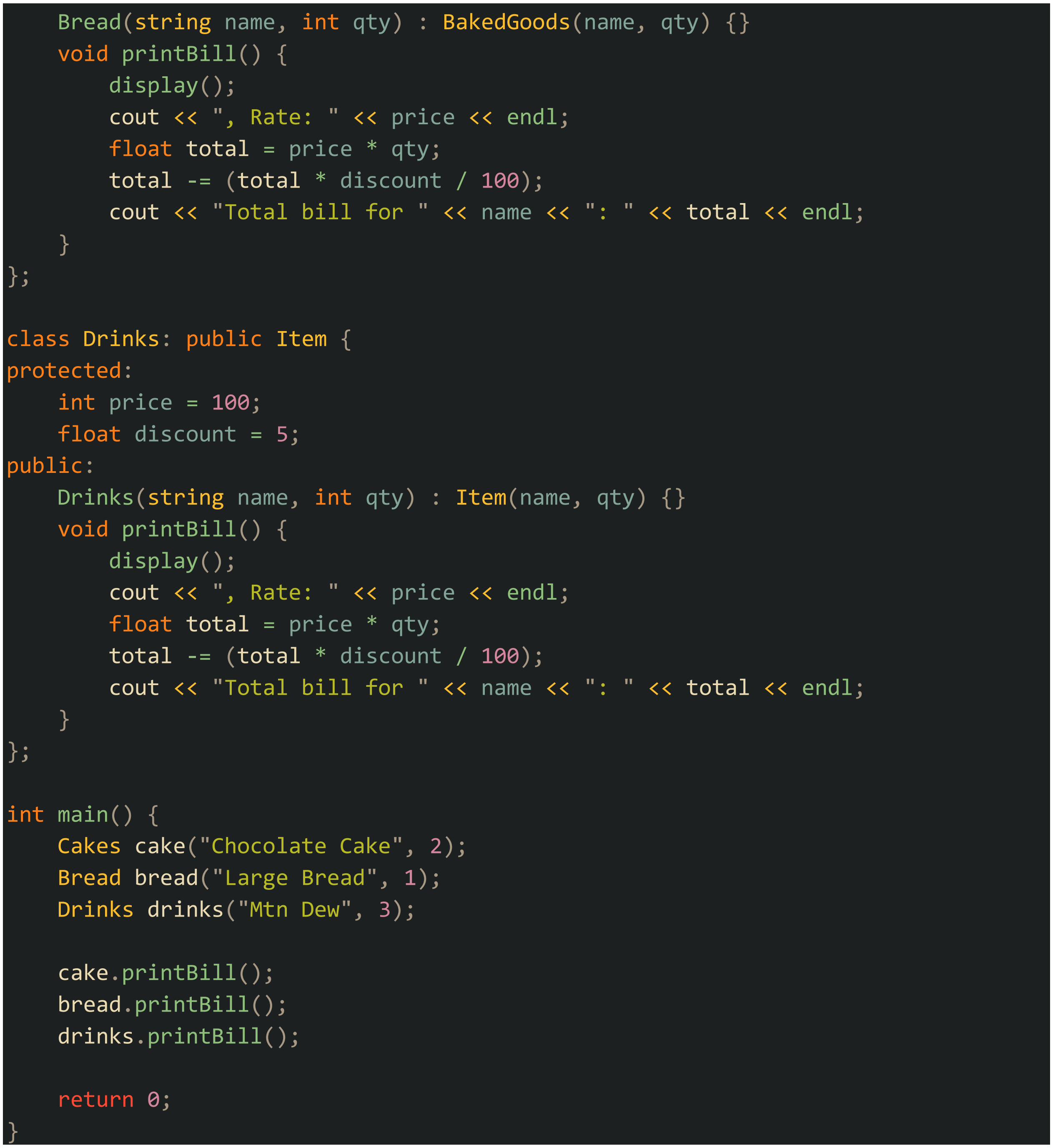


```
Weapons are used for DESTRUCTION!!!
Hot Weapons uses Gun Powder!!!
Bombs blow up!!!
Nuclear bombs blow up and use nuclear fission and fusion!!!
```

QUESTION#4:

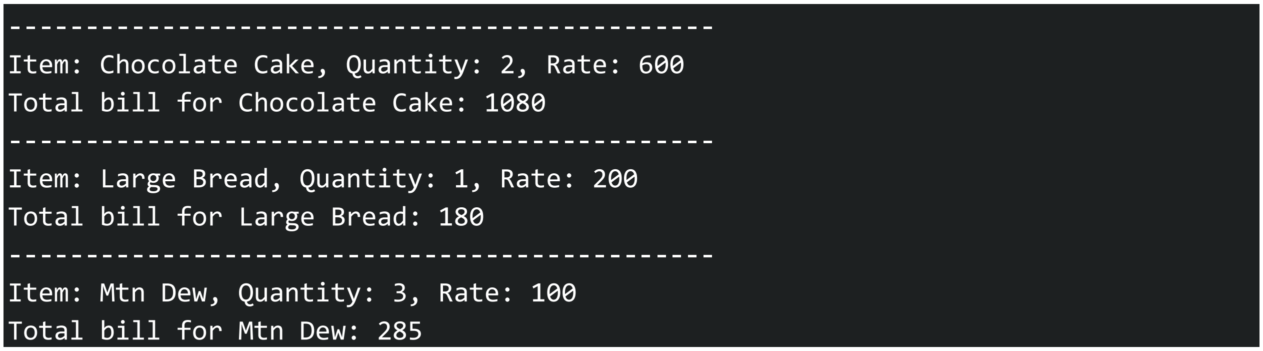
CODE:

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

class Item {
protected:
    string name;
    int qty;
public:
    Item(string name, int qty) : name(name), qty(qty) {}
    void display() {
        cout << "----------------------------------------------\n";
        cout << "Item: " << name << ", Quantity: " << qty;
    }
};

class BakedGoods: public Item {
protected:
    float discount = 10;
public:
    BakedGoods(string name, int qty) : Item(name, qty) {}
};

class Cakes: public BakedGoods {
protected:
    int price = 600;
public:
    Cakes(string name, int qty) : BakedGoods(name, qty) {}
    void printBill() {
        display();
        cout << ", Rate: " << price << endl;
        float total = price * qty;
        total -= (total * discount / 100);
        cout << "Total bill for " << name << ": " << total << endl;
    }
};

class Bread: public BakedGoods {
protected:
    int price = 200;
public:
```



```
    Bread(string name, int qty) : BakedGoods(name, qty) {}
    void printBill() {
        display();
        cout << ", Rate: " << price << endl;
        float total = price * qty;
        total -= (total * discount / 100);
        cout << "Total bill for " << name << ": " << total << endl;
    }
};

class Drinks: public Item {
protected:
    int price = 100;
    float discount = 5;
public:
    Drinks(string name, int qty) : Item(name, qty) {}
    void printBill() {
        display();
        cout << ", Rate: " << price << endl;
        float total = price * qty;
        total -= (total * discount / 100);
        cout << "Total bill for " << name << ": " << total << endl;
    }
};

int main() {
    Cakes cake("Chocolate Cake", 2);
    Bread bread("Large Bread", 1);
    Drinks drinks("Mtn Dew", 3);

    cake.printBill();
    bread.printBill();
    drinks.printBill();

    return 0;
}
```

OUTPUT:



```
----------------------------------------------
Item: Chocolate Cake, Quantity: 2, Rate: 600
Total bill for Chocolate Cake: 1080
----------------------------------------------
Item: Large Bread, Quantity: 1, Rate: 200
Total bill for Large Bread: 180
----------------------------------------------
Item: Mtn Dew, Quantity: 3, Rate: 100
Total bill for Mtn Dew: 285
```