INTERNATIONAL ISLAMIC UNIVERSITY CHITTAGONG CSE-1222

COMPUTER PROGRAMMING 2.

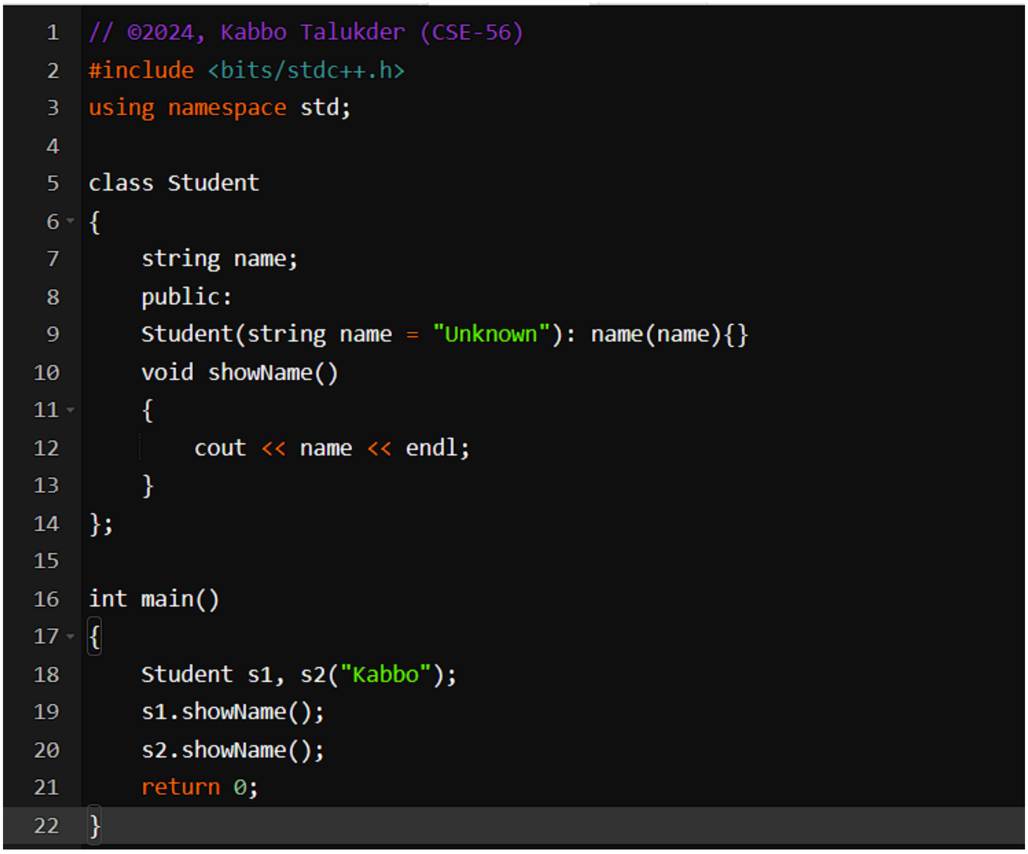
LAB LAB PRACTICE SHEET 2

See Details: C++ Class

Segment Pdf: All The Segment of c++

6. Write a program to print the names of students by creating a Student class. If no name is passed while creating an object of the Student class, then the name should be "Unknown", otherwise the name should be equal to the String value passed while creating the object of the Student class

Solve:



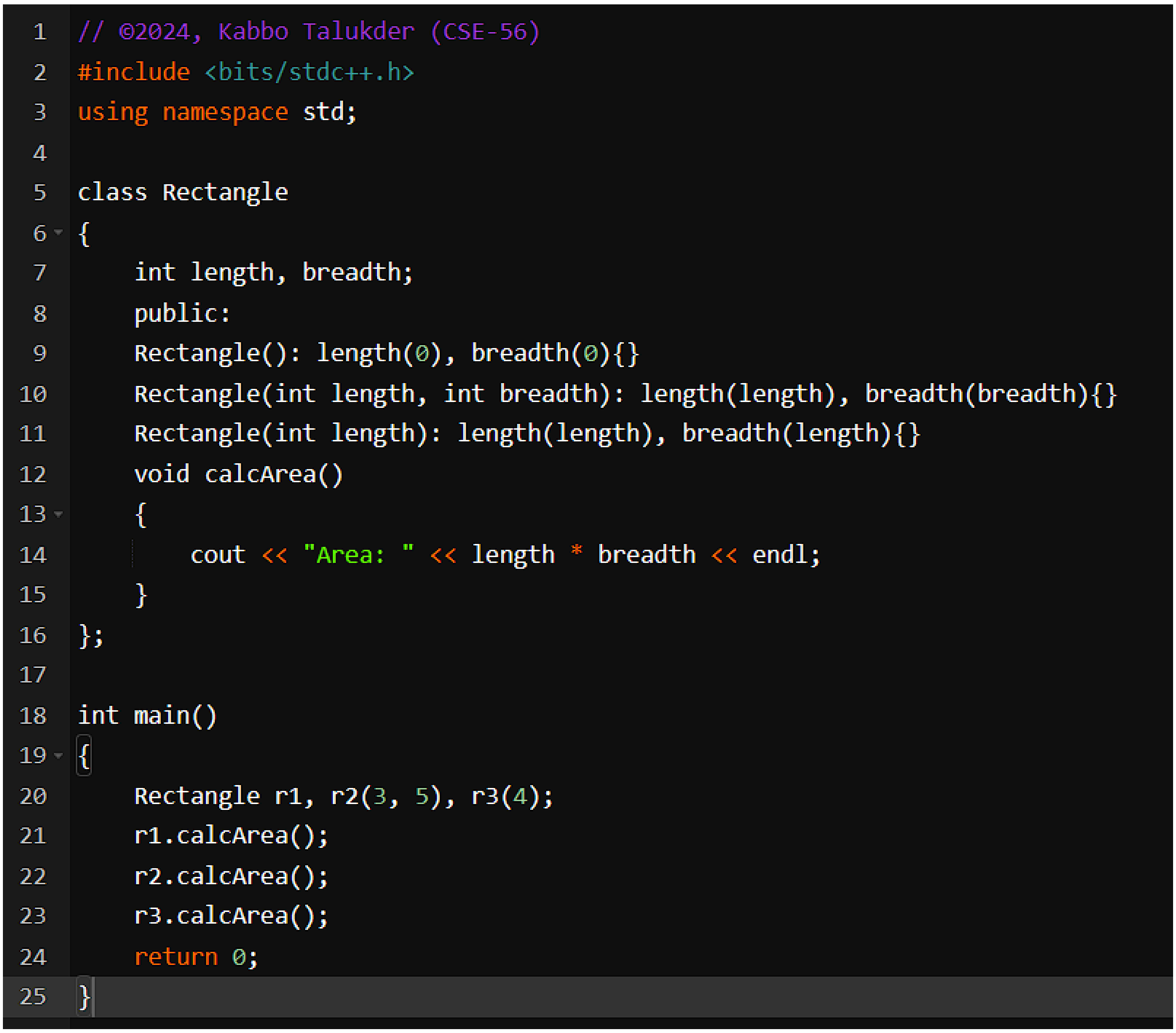
```cpp
// ©2024, Kabbo Talukder (CSE-56)
#include <bits/stdc++.h>
using namespace std;

class Student
{
    string name;
    public:
    Student(string name = "Unknown"): name(name){}
    void showName()
    {
        cout << name << endl;
    }
};

int main()
{
    Student s1, s2("Kabbo");
    s1.showName();
    s2.showName();
    return 0;
}
```

7. Create a class named 'Rectangle' with two data members- length and breadth and a function to calculate the area which is 'length*breadth'. The class has three constructors which are:

1 - having no parameter - values of both length and breadth are assigned zero.

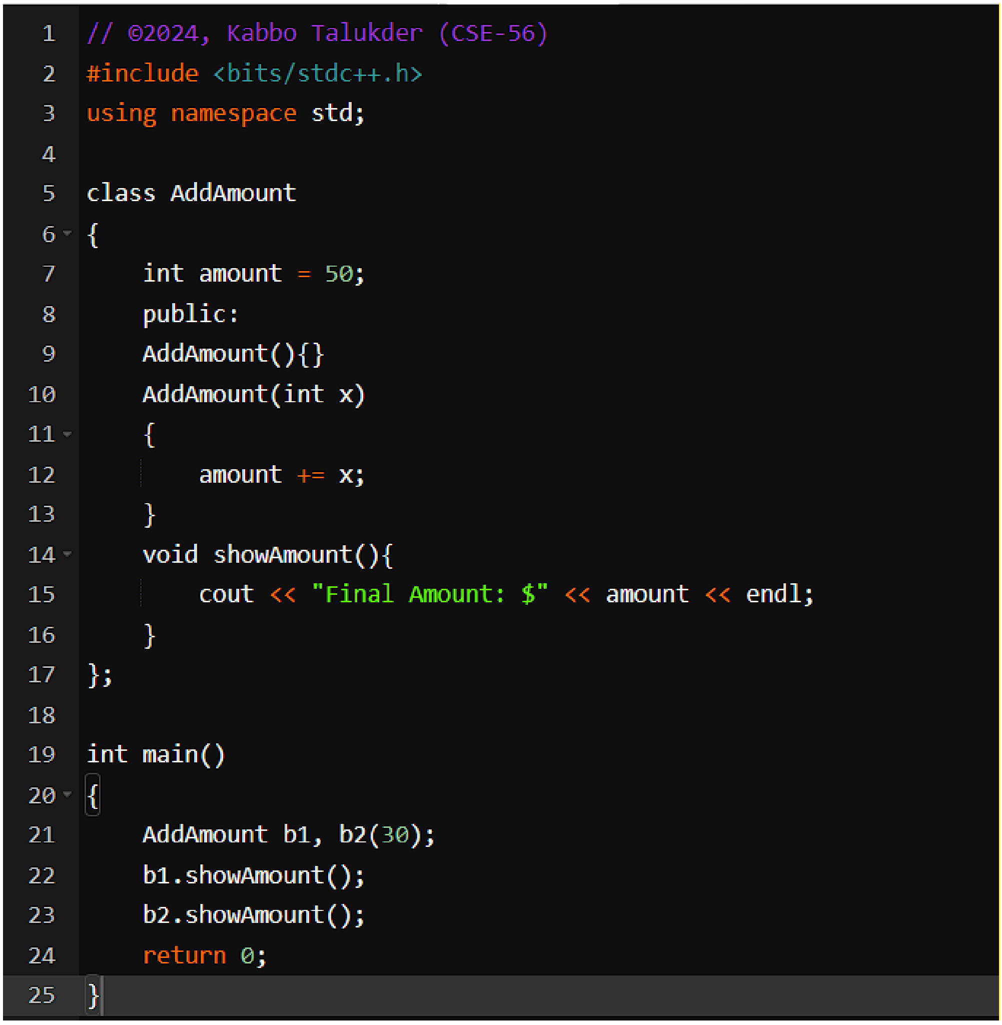
2 - Having two numbers as parameters - the two numbers are assigned as length and breadth respectively.

3 - Having one number as parameter - both length and breadth are assigned that number. Now, create objects of the 'Rectangle' class having none, one and two parameters and print their areas.



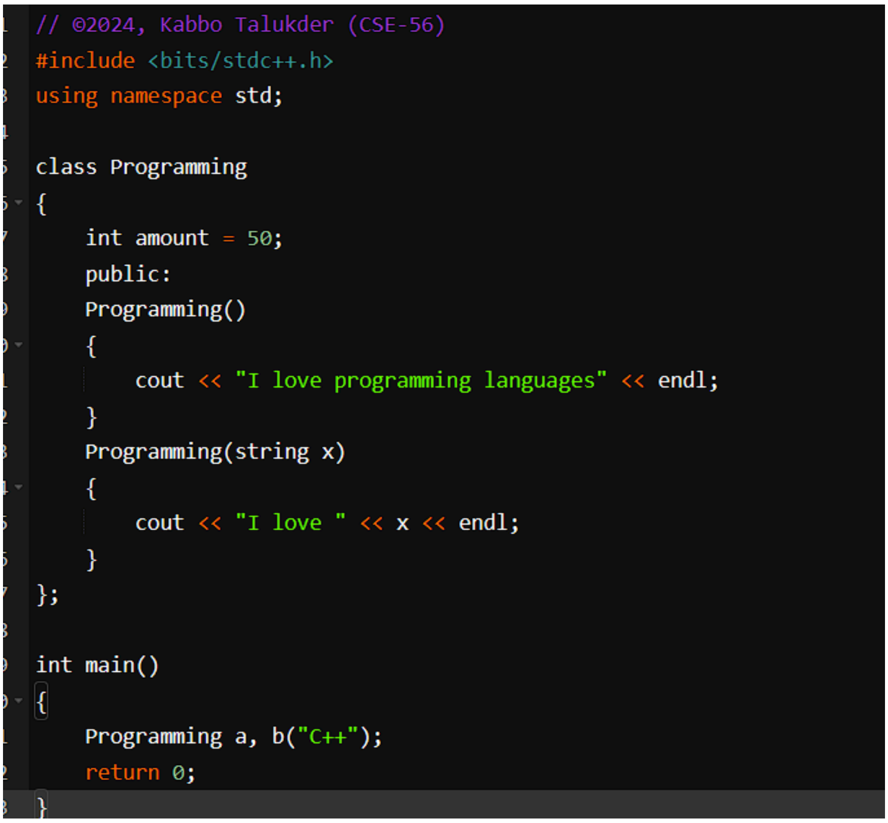
```cpp
// ©2024, Kabbo Talukder (CSE-56)
#include <bits/stdc++.h>
using namespace std;

class Rectangle
{
    int length, breadth;
    public:
    Rectangle(): length(0), breadth(0){}
    Rectangle(int length, int breadth): length(length), breadth(breadth){}
    Rectangle(int length): length(length), breadth(length){}
    void calcArea()
    {
        cout << "Area: " << length * breadth << endl;
    }
};

int main()
{
    Rectangle r1, r2(3, 5), r3(4);
    r1.calcArea();
    r2.calcArea();
    r3.calcArea();
    return 0;
}
```

8. Suppose you have a Bank with an initial amount of $50 and you have to add some more amount to it. Create a class 'AddAmount' with a data member named 'amount' with an initial value of $50. Now make two constructors of this class as follows: a. without any parameter - no amount will be added to the Bank b. Having a parameter which is the amount that will be added to the Bank Create an object of the 'AddAmount' class and display the final amount in the Bank.



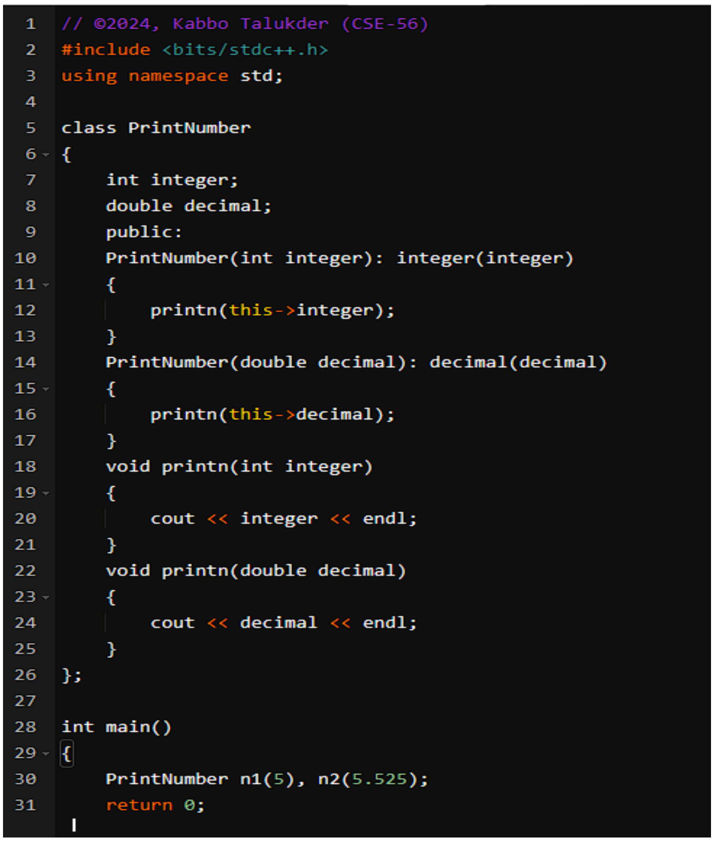
```cpp
// ©2024, Kabbo Talukder (CSE-56)
#include <bits/stdc++.h>
using namespace std;

class AddAmount
{
    int amount = 50;
    public:
    AddAmount(){}
    AddAmount(int x)
    {
        amount += x;
    }
    void showAmount(){
        cout << "Final Amount: $" << amount << endl;
    }
};

int main()
{
    AddAmount b1, b2(30);
    b1.showAmount();
    b2.showAmount();
    return 0;
}
```

9. Create a class named 'Programming'. While creating an object of the class, if nothing is passed to it, then the message "I love programming languages" should be printed. If some String is passed to it, then in place of "programming languages" the name of that String variable should be printed. For example, while creating the object if we pass "cpp", then "I love cpp" should be printed.



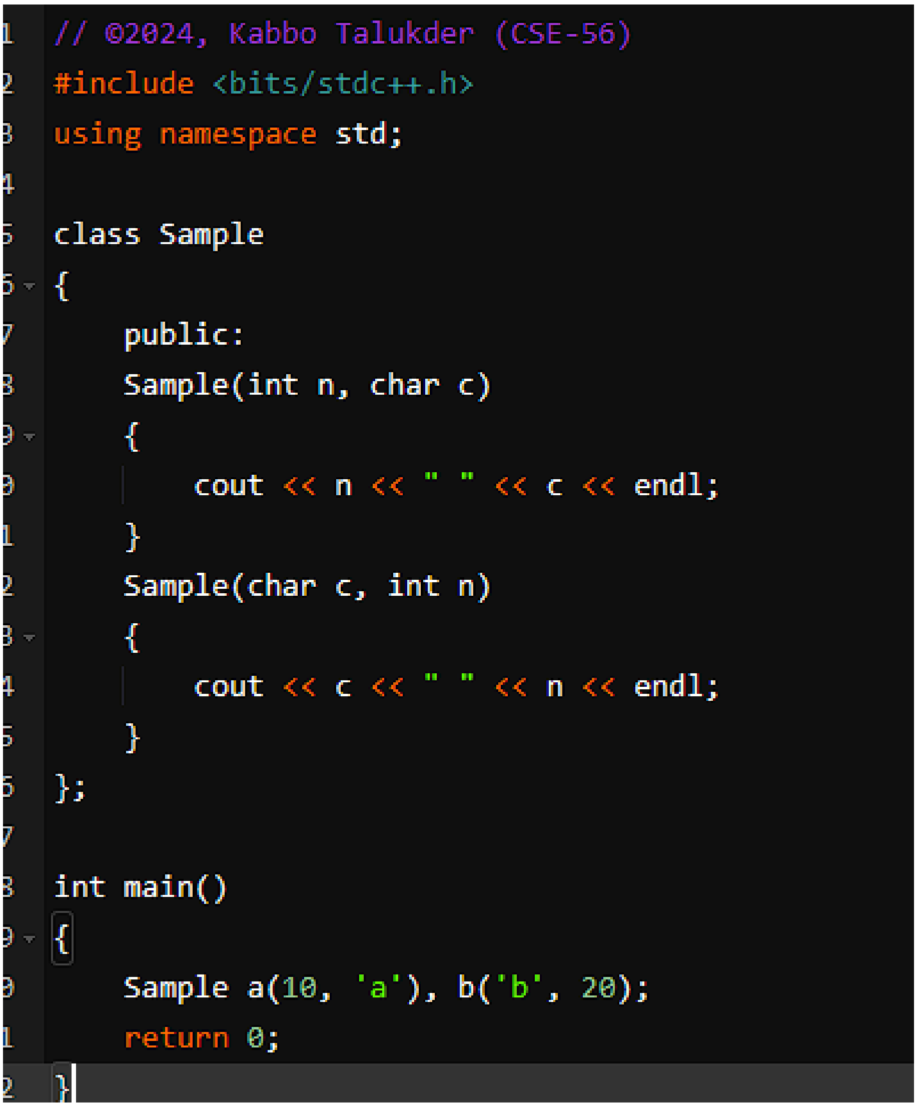
```cpp
// ©2024, Kabbo Talukder (CSE-56)
#include <bits/stdc++.h>
using namespace std;

class Programming
{
    int amount = 50;
    public:
    Programming()
    {
        cout << "I love programming languages" << endl;
    }
    Programming(string x)
    {
        cout << "I love " << x << endl;
    }
};

int main()
{
    Programming a, b("C++");
    return 0;
}
```

10. Create a class named 'PrintNumber' to print various numbers of different datatypes by creating different functions with the same name 'printn' having a parameter for each datatype.



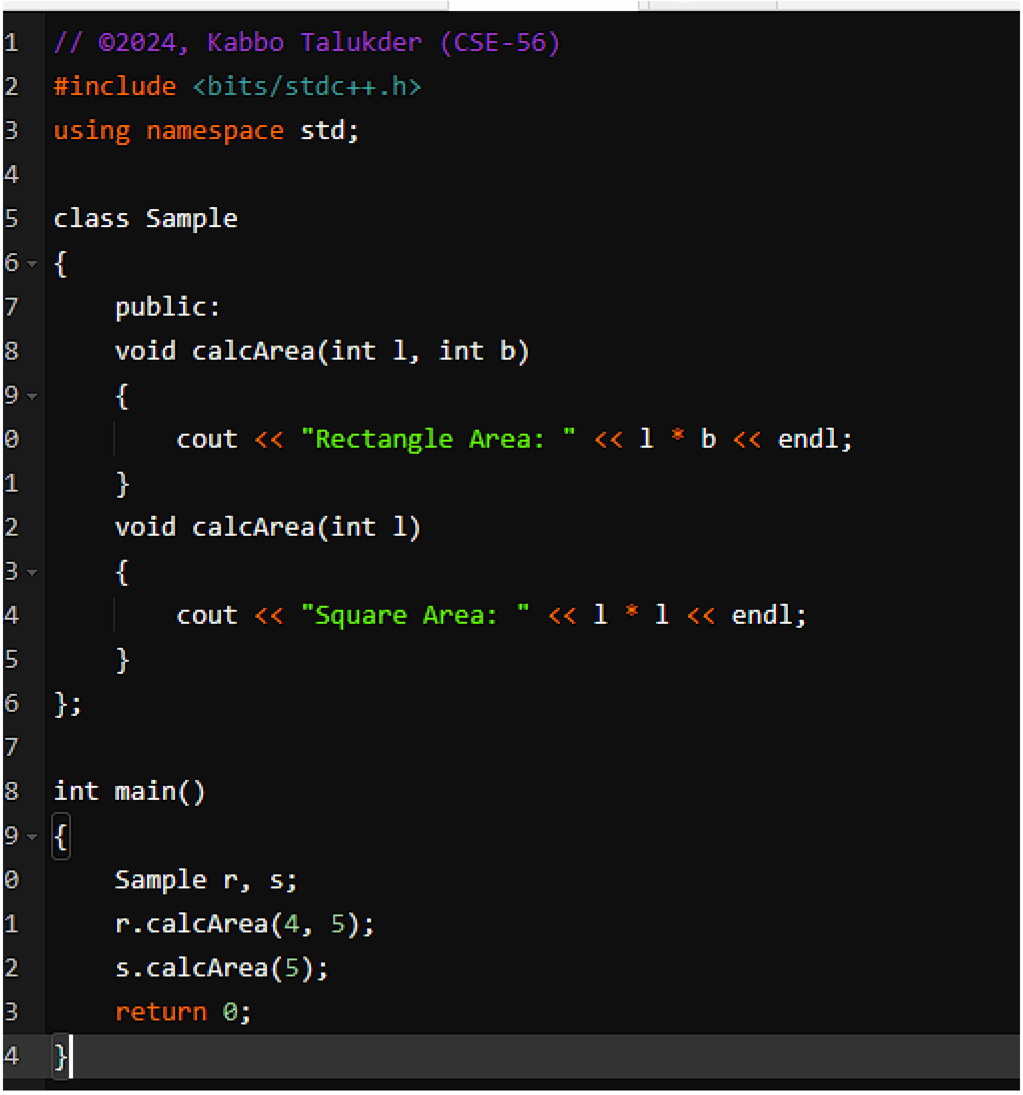
```cpp
// ©2024, Kabbo Talukder (CSE-56)
#include <bits/stdc++.h>
using namespace std;

class PrintNumber
{
    int integer;
    double decimal;
    public:
    PrintNumber(int integer): integer(integer)
    {
        printn(this->integer);
    }
    PrintNumber(double decimal): decimal(decimal)
    {
        printn(this->decimal);
    }
    void printn(int integer)
    {
        cout << integer << endl;
    }
    void printn(double decimal)
    {
        cout << decimal << endl;
    }
};

int main()
{
    PrintNumber n1(5), n2(5.525);
    return 0;
```

11. Create a class to print an integer and a character using two functions having the same name but different sequence of the integer and the character parameters. For example, if the parameters of the first function are of the form (int n, char c), then that of the second function will be of the form (char c, int n)



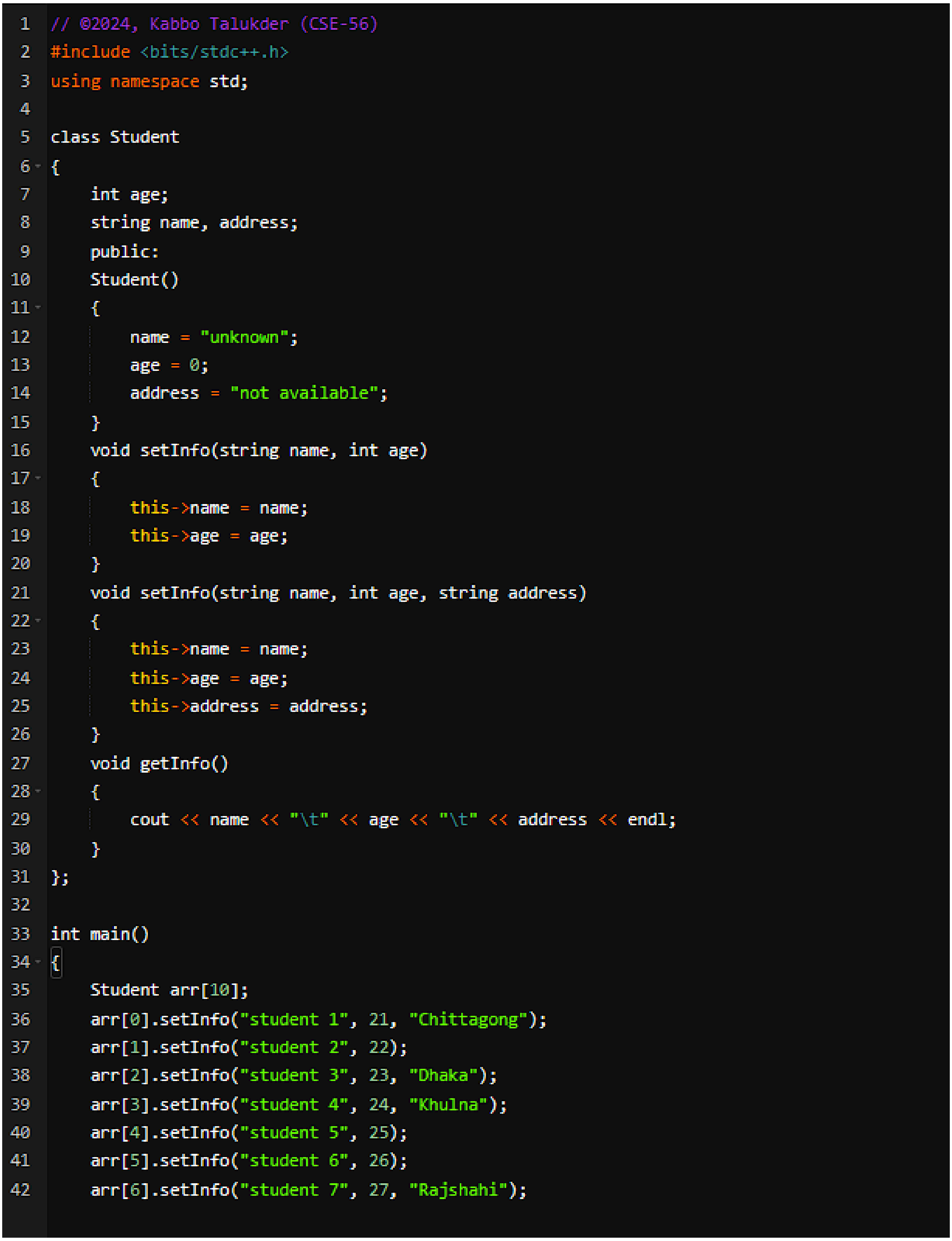
```cpp
// ©2024, Kabbo Talukder (CSE-56)
#include <bits/stdc++.h>
using namespace std;

class Sample
{
    public:
    Sample(int n, char c)
    {
        cout << n << " " << c << endl;
    }
    Sample(char c, int n)
    {
        cout << c << " " << n << endl;
    }
};

int main()
{
    Sample a(10, 'a'), b('b', 20);
    return 0;
}
```

12. Create a class to print the area of a square and a rectangle. The class has two functions with the same name but different number of parameters. The function for printing the area of rectangle has two parameters which are its length and breadth respectively while the other function for printing the area of square has one parameter which is the side of the square.



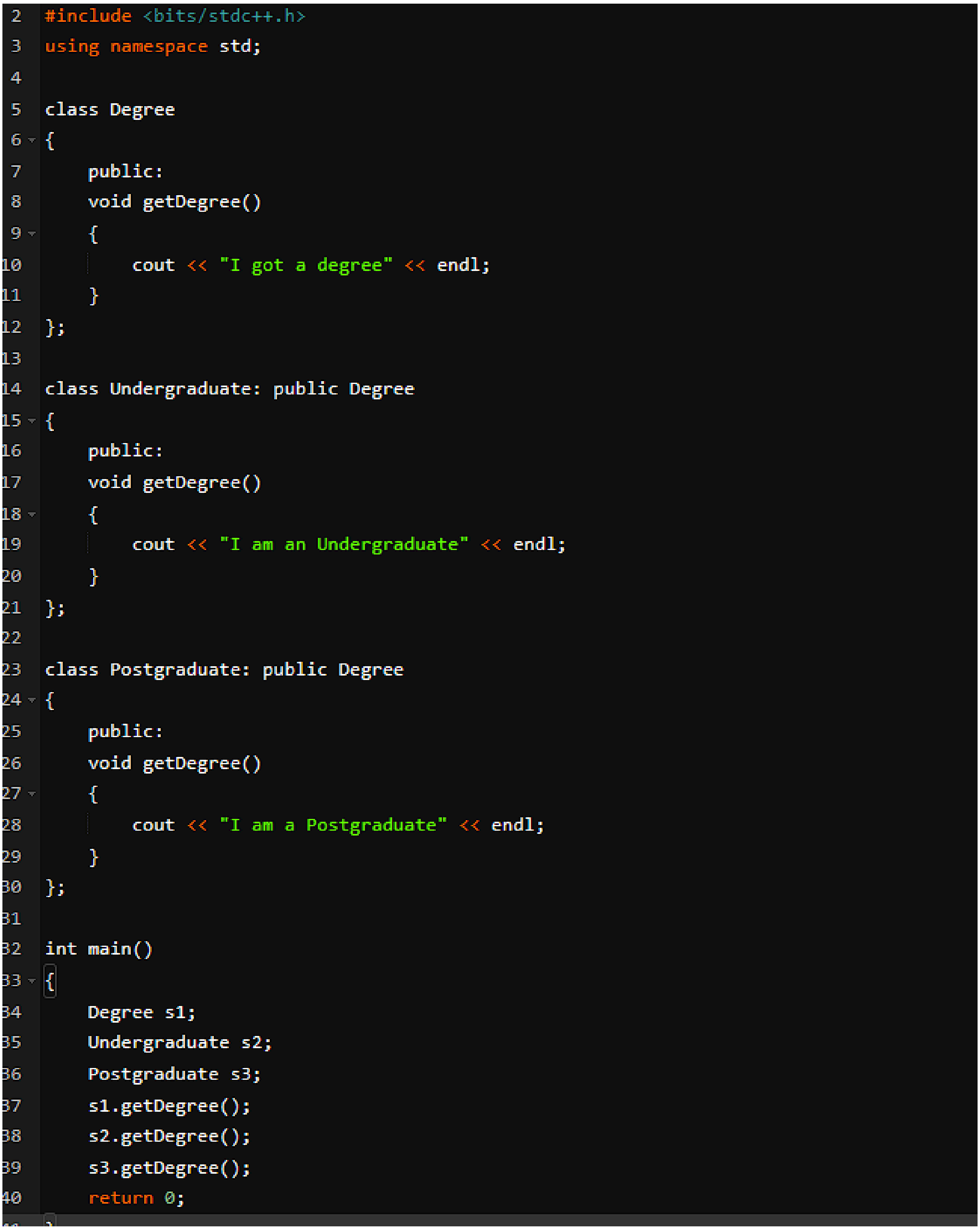
```cpp
// ©2024, Kabbo Talukder (CSE-56)
#include <bits/stdc++.h>
using namespace std;

class Sample
{
    public:
    void calcArea(int l, int b)
    {
        cout << "Rectangle Area: " << l * b << endl;
    }
    void calcArea(int l)
    {
        cout << "Square Area: " << l * l << endl;
    }
};

int main()
{
    Sample r, s;
    r.calcArea(4, 5);
    s.calcArea(5);
    return 0;
}
```

13. Create a class 'Student' with three data members which are name, age and address. The constructor of the class assigns default values to name as "unknown", age as '0' and address as "not available". It has two functions with the same name 'setInfo'. First function has two parameters for name and age and assigns the same whereas the second function takes has three parameters which are assigned to name, age and address respectively. Print the name, age and address of 10 students. [Hint - Use array of objects]



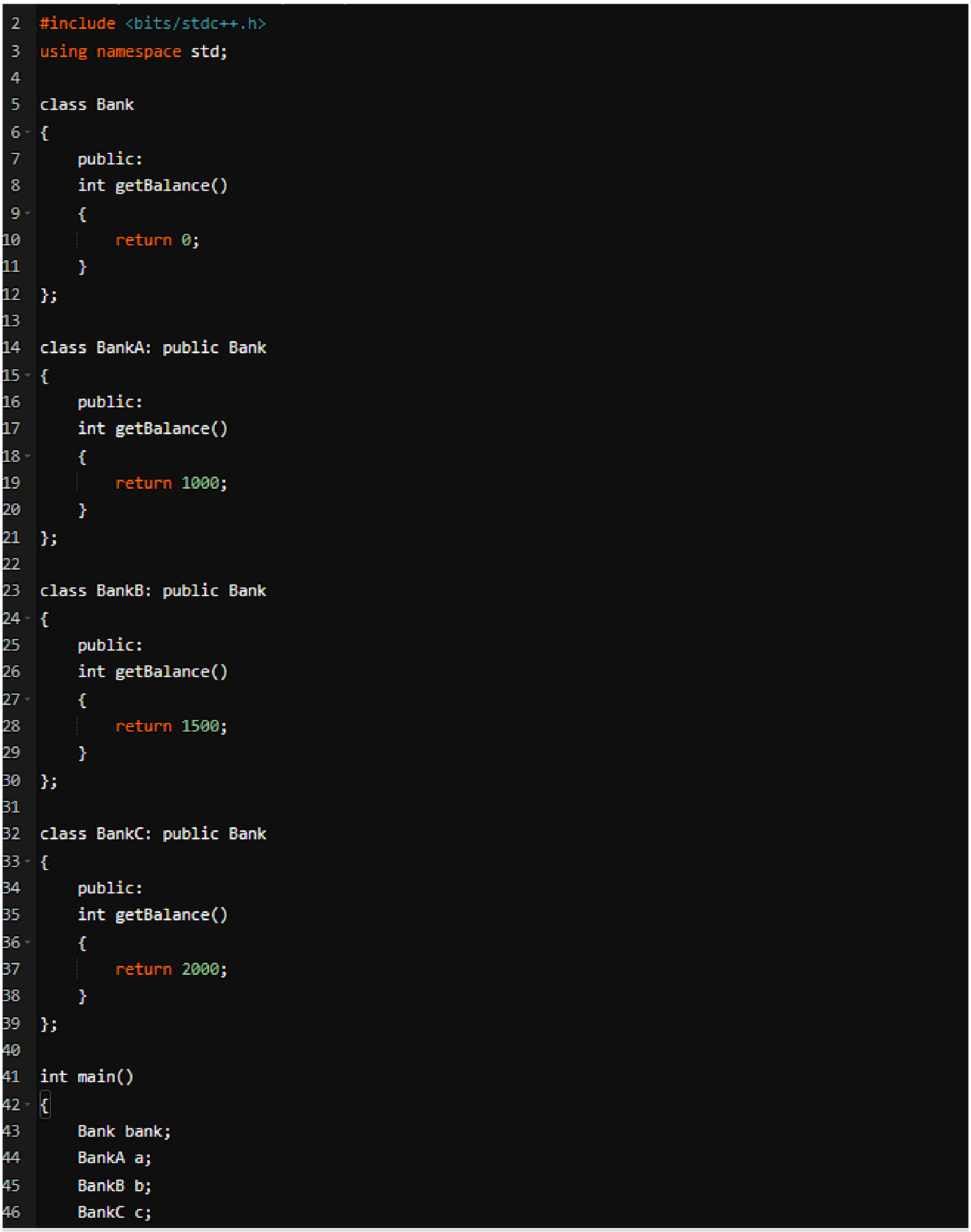
```cpp
// ©2024, Kabbo Talukder (CSE-56)
#include <bits/stdc++.h>
using namespace std;

class Student
{
    int age;
    string name, address;
    public:
    Student()
    {
        name = "unknown";
        age = 0;
        address = "not available";
    }
    void setInfo(string name, int age)
    {
        this->name = name;
        this->age = age;
    }
    void setInfo(string name, int age, string address)
    {
        this->name = name;
        this->age = age;
        this->address = address;
    }
    void getInfo()
    {
        cout << name << "\t" << age << "\t" << address << endl;
    }
};

int main()
{
    Student arr[10];
    arr[0].setInfo("student 1", 21, "Chittagong");
    arr[1].setInfo("student 2", 22);
    arr[2].setInfo("student 3", 23, "Dhaka");
    arr[3].setInfo("student 4", 24, "Khulna");
    arr[4].setInfo("student 5", 25);
    arr[5].setInfo("student 6", 26);
    arr[6].setInfo("student 7", 27, "Rajshahi");
```

14. Create a class 'Degree' having a function 'getDegree' that prints "I got a degree". It has two subclasses namely 'Undergraduate' and 'Postgraduate' each having a function with the same name that prints "I am an Undergraduate" and "I am a Postgraduate" respectively. Call the function by creating an object of each of the three classes.



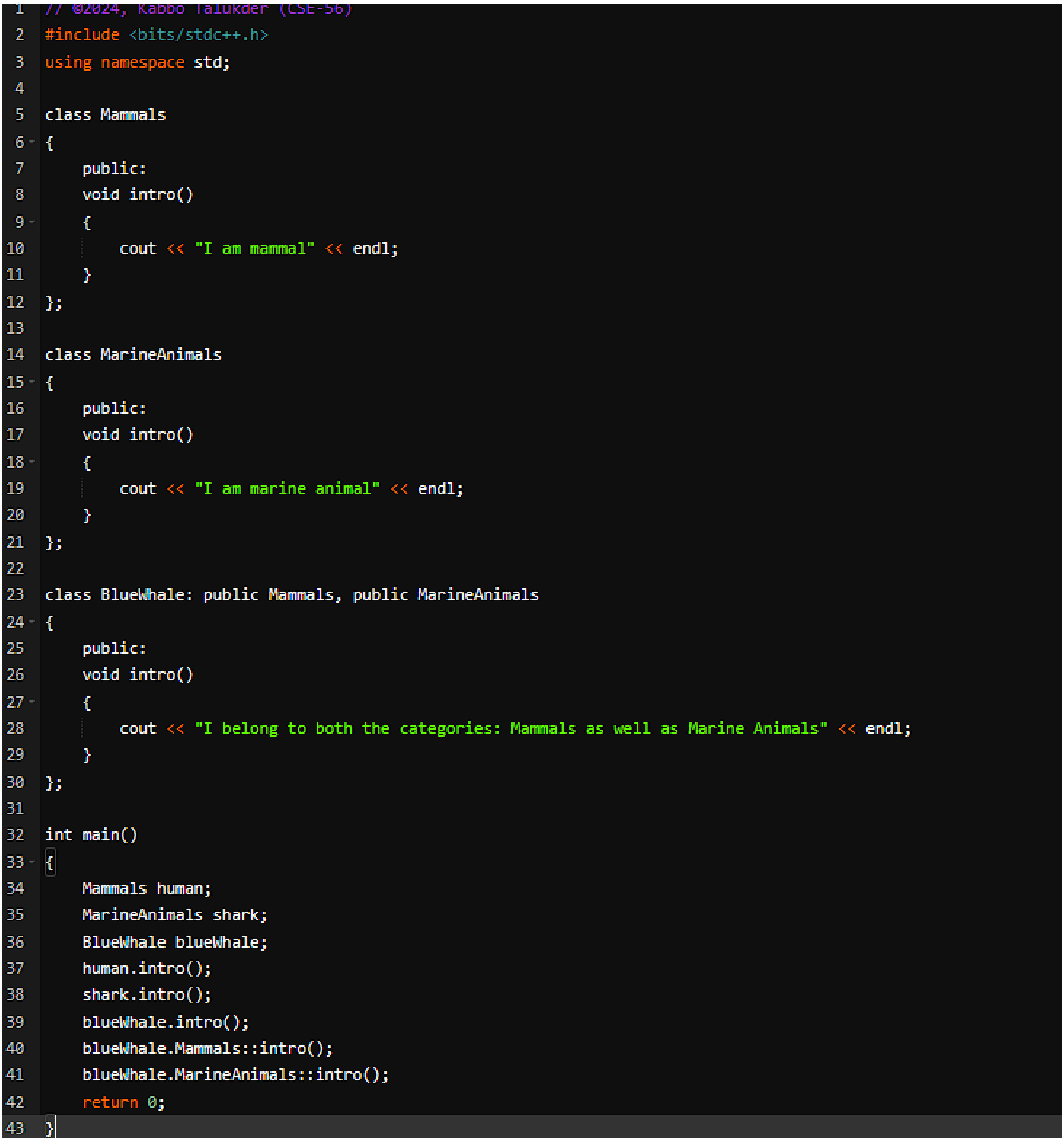
```
#include <bits/stdc++.h>
using namespace std;

class Degree
{
    public:
    void getDegree()
    {
        cout << "I got a degree" << endl;
    }
};

class Undergraduate: public Degree
{
    public:
    void getDegree()
    {
        cout << "I am an Undergraduate" << endl;
    }
};

class Postgraduate: public Degree
{
    public:
    void getDegree()
    {
        cout << "I am a Postgraduate" << endl;
    }
};

int main()
{
    Degree s1;
    Undergraduate s2;
    Postgraduate s3;
    s1.getDegree();
    s2.getDegree();
    s3.getDegree();
    return 0;
```

15. A boy has his money deposited $1000, $1500 and $2000 in banks-Bank A, Bank B and Bank C respectively. We have to print the money deposited by him in a particular bank.Create a class 'Bank' with a function 'getBalance' which returns 0. Make its three subclasses named 'BankA', 'BankB' and 'BankC' with a function with the same name 'getBalance' which returns the amount deposited in that particular bank. Call the function 'getBalance' by the object of each of the three banks.



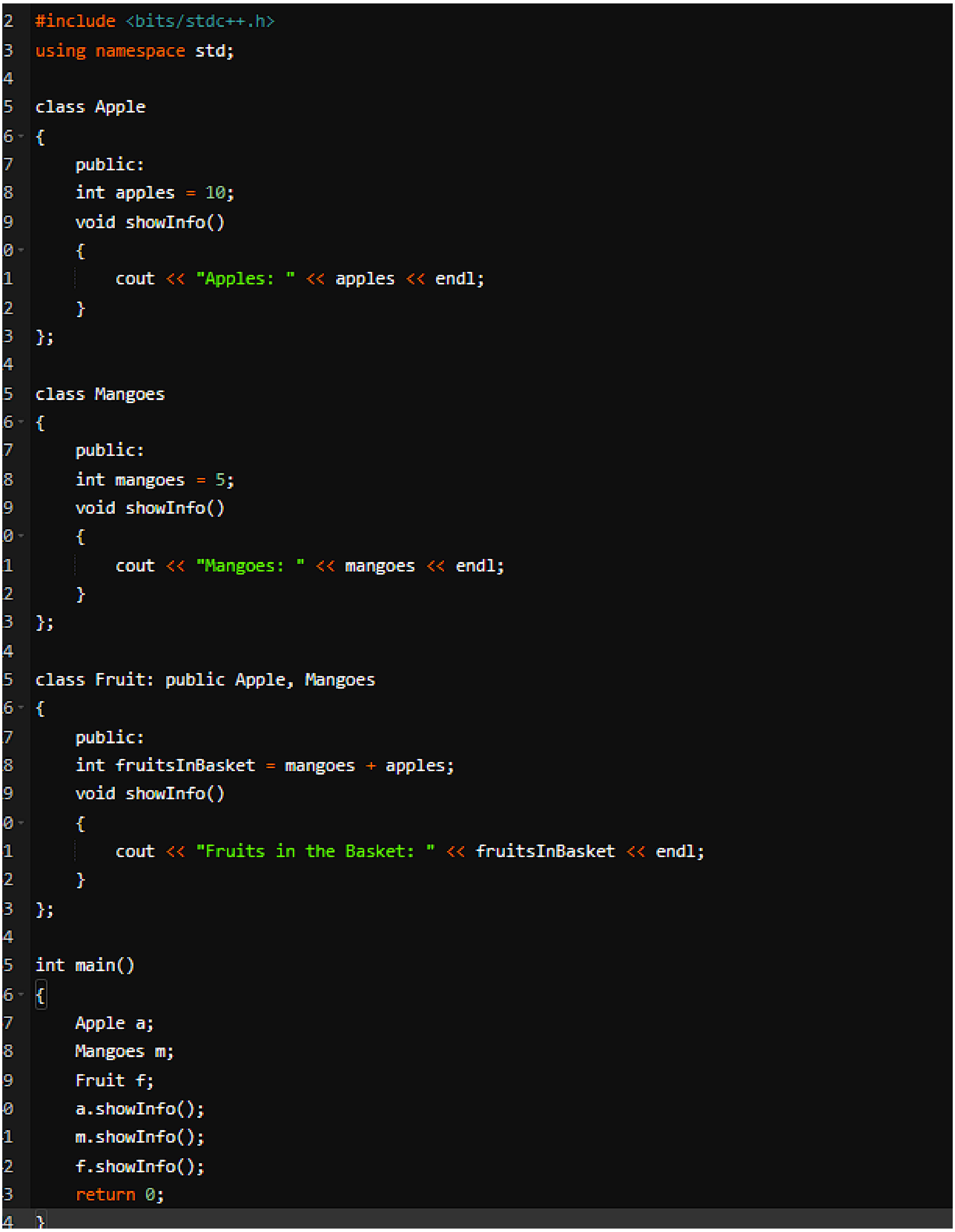
```
#include <bits/stdc++.h>
using namespace std;

class Bank
{
    public:
    int getBalance()
    {
        return 0;
    }
};

class BankA: public Bank
{
    public:
    int getBalance()
    {
        return 1000;
    }
};

class BankB: public Bank
{
    public:
    int getBalance()
    {
        return 1500;
    }
};

class BankC: public Bank
{
    public:
    int getBalance()
    {
        return 2000;
    }
};

int main()
{
    Bank bank;
    BankA a;
    BankB b;
    BankC c;
```

19. Create two classes named Mammals and MarineAnimals. Create another class named BlueWhale which inherits both the above classes. Now, create a function in each of these classes which prints "I am mammal", "I am a marine animal" and "I belong to both the categories: Mammals as well as Marine Animals" respectively. Now, create an object for each of the above class and try calling a. function of Mammals by the object of Mammal b. of MarineAnimal by the object of MarineAnimal c. function of BlueWhale by the object of BlueWhale d. function of each of its parent by the object of BlueWhale



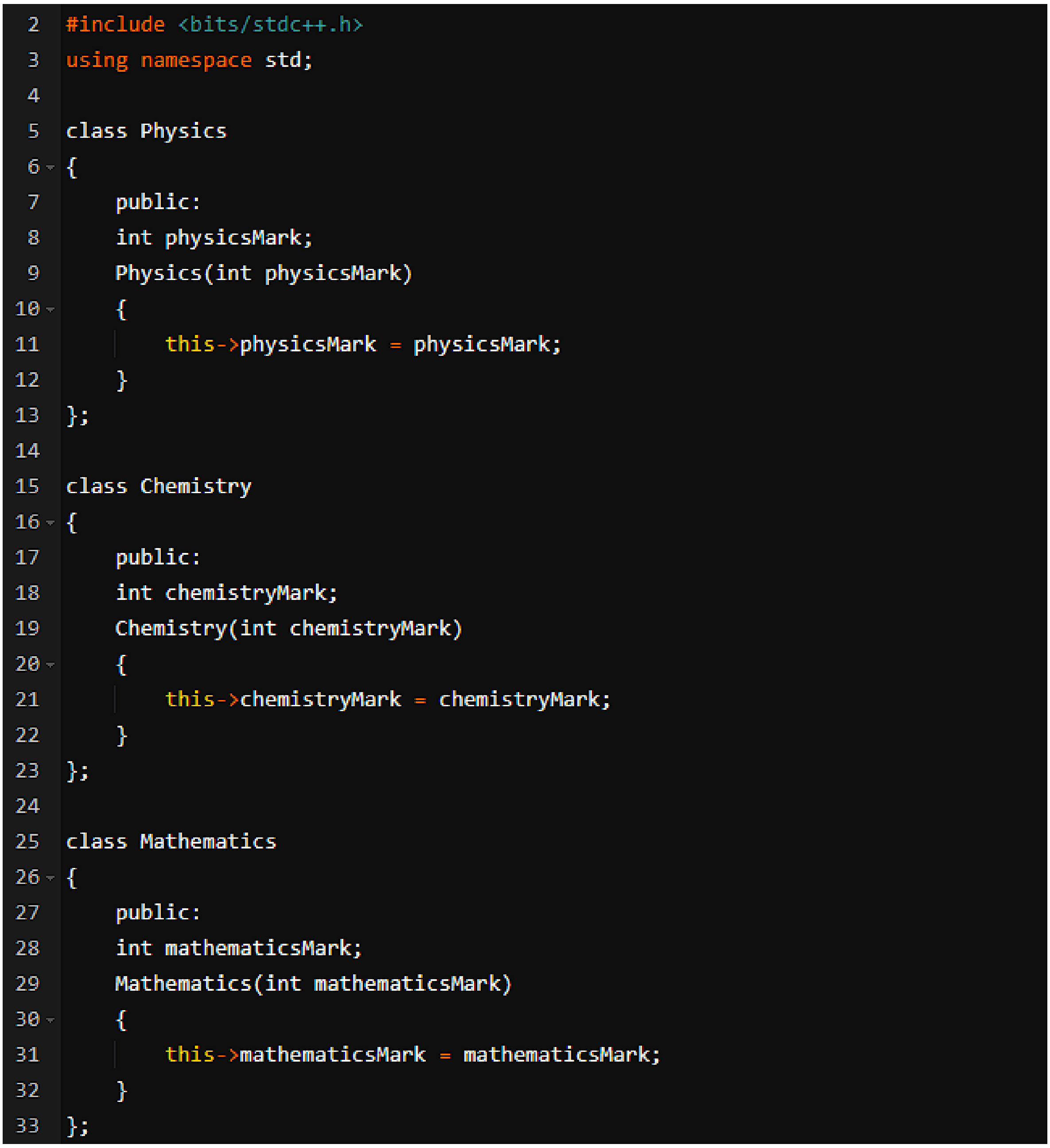
```cpp
// ©2024, Kabbo Talukder (CSE-56)
#include <bits/stdc++.h>
using namespace std;

class Mammals
{
    public:
    void intro()
    {
        cout << "I am mammal" << endl;
    }
};

class MarineAnimals
{
    public:
    void intro()
    {
        cout << "I am marine animal" << endl;
    }
};

class BlueWhale: public Mammals, public MarineAnimals
{
    public:
    void intro()
    {
        cout << "I belong to both the categories: Mammals as well as Marine Animals" << endl;
    }
};

int main()
{
    Mammals human;
    MarineAnimals shark;
    BlueWhale blueWhale;
    human.intro();
    shark.intro();
    blueWhale.intro();
    blueWhale.Mammals::intro();
    blueWhale.MarineAnimals::intro();
    return 0;
}
```

20. Make a class named Fruit with a data member to calculate the number of fruits in a basket. Create two other class named Apples and Mangoes to calculate the number of apples and mangoes in the basket. Print the number of fruits of each type and the total number of fruits in the basket.

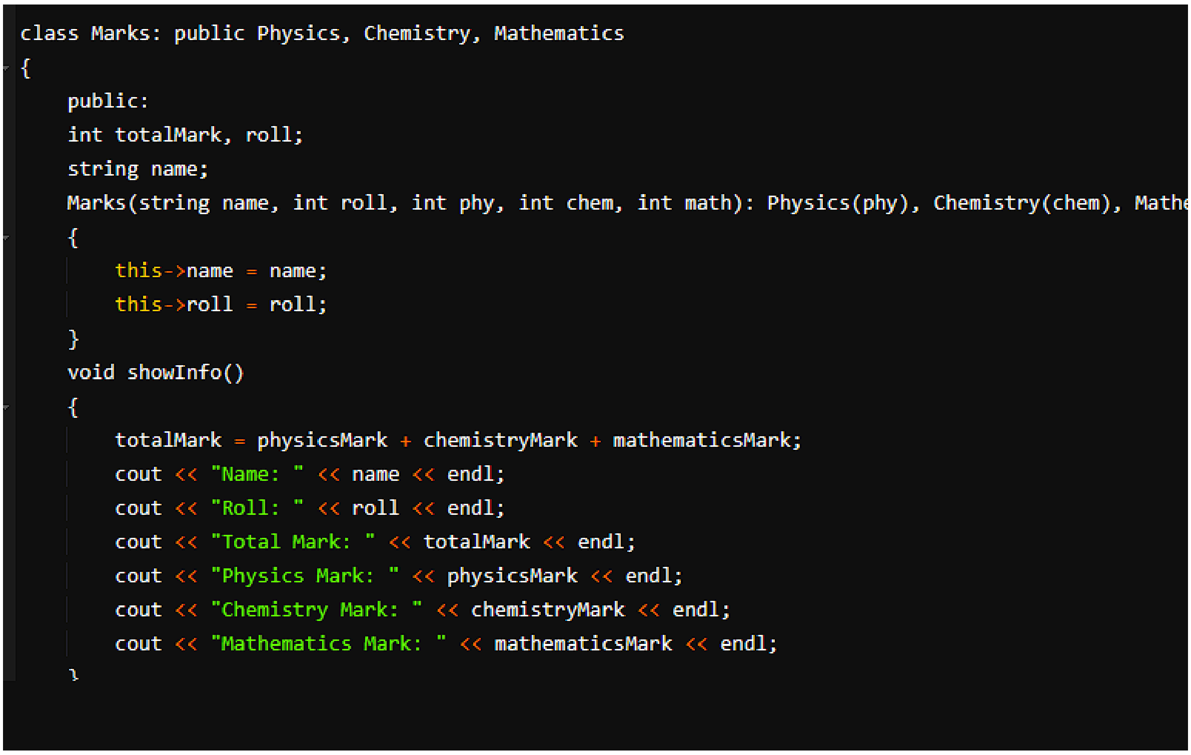


```
#include <bits/stdc++.h>
using namespace std;

class Apple
{
    public:
    int apples = 10;
    void showInfo()
    {
        cout << "Apples: " << apples << endl;
    }
};

class Mangoes
{
    public:
    int mangoes = 5;
    void showInfo()
    {
        cout << "Mangoes: " << mangoes << endl;
    }
};

class Fruit: public Apple, Mangoes
{
    public:
    int fruitsInBasket = mangoes + apples;
    void showInfo()
    {
        cout << "Fruits in the Basket: " << fruitsInBasket << endl;
    }
};

int main()
{
    Apple a;
    Mangoes m;
    Fruit f;
    a.showInfo();
    m.showInfo();
    f.showInfo();
    return 0;
}
```

21. We want to calculate the total marks of each student of a class in Physics, Chemistry and Mathematics and the average marks of the class. The number of students in the class are entered by the user. Create a class named Marks with data members for roll number, name and marks. Create three other classes inheriting the Marks class, namely Physics, Chemistry and Mathematics, which are used to define marks in individual subject of each student. Roll number of each student will be generated automatically.



```cpp
#include <bits/stdc++.h>
using namespace std;

class Physics
{
    public:
    int physicsMark;
    Physics(int physicsMark)
    {
        this->physicsMark = physicsMark;
    }
};

class Chemistry
{
    public:
    int chemistryMark;
    Chemistry(int chemistryMark)
    {
        this->chemistryMark = chemistryMark;
    }
};

class Mathematics
{
    public:
    int mathematicsMark;
    Mathematics(int mathematicsMark)
    {
        this->mathematicsMark = mathematicsMark;
    }
};
```



```
class Marks: public Physics, Chemistry, Mathematics
{
    public:
    int totalMark, roll;
    string name;
    Marks(string name, int roll, int phy, int chem, int math): Physics(phy), Chemistry(chem), Mathe
    {
        this->name = name;
        this->roll = roll;
    }
    void showInfo()
    {
        totalMark = physicsMark + chemistryMark + mathematicsMark;
        cout << "Name: " << name << endl;
        cout << "Roll: " << roll << endl;
        cout << "Total Mark: " << totalMark << endl;
        cout << "Physics Mark: " << physicsMark << endl;
        cout << "Chemistry Mark: " << chemistryMark << endl;
        cout << "Mathematics Mark: " << mathematicsMark << endl;
    }
```

22. We want to store the information of different vehicles. Create a class named Vehicle with two data member named mileage and price. Create its two subclasses *Car with data members to store ownership cost, warranty (by years), seating capacity and fuel type (diesel or petrol). *Bike with data members to store the number of cylinders, number of gears, cooling type(air, liquid or oil), wheel type(alloys or spokes) and fuel tank size(in inches) Make another two subclasses Audi and Ford of Car, each having a data member to store the model type. Next, make two subclasses Walton and TVS, each having a data member to store the make-type. Now, store and print the information of an Audi and a Ford car (i.e. model type, ownership cost, warranty, seating capacity, fuel type, mileage and price.) Do the same for a Walton and a TVS bike

```
// ©2024, Kabbo Talukder (CSE-56)
#include <bits/stdc++.h>
using namespace std;
class Vehicle
{
    public:
    double milage;
    int price;
    Vehicle(double milage, int price): milage(milage), price(price){}
};
class Car: public Vehicle
{
    public:
    int storeOwnershipCost, warranty, seatingCapacity;
    string fuelType;
    Car(int storeOwnershipCost, int warranty, int seatingCapacity, string fuelType,
double milage, int price):
        Vehicle(milage, price), storeOwnershipCost(storeOwnershipCost),
warranty(warranty), seatingCapacity(seatingCapacity), fuelType(fuelType){}
};
class Bike: public Vehicle
{
    public:
    int numberOfCylinders, numberOfGears, fuelTankSize;
    string coolingType, wheelType;
};
class Audi: public Car
{
    public:
    string modelType;
    Audi(string modelType, int storeOwnershipCost, int warranty, int
seatingCapacity, string fuelType , double milage, int price):
    Car(storeOwnershipCost, warranty, seatingCapacity, fuelType, milage, price),
modelType(modelType){}
    void showInfo()
    {
        cout << "Model Type: " << modelType << endl;
        cout << "Store Ownership Cost: " << storeOwnershipCost << " Taka" <<
endl;
        cout << "Warranty: " << warranty << " Year" << endl;
        cout << "Seating Capacity: " << seatingCapacity << endl;
        cout << "Fuel Type: " << fuelType << endl;
        cout << "Milage: " << milage << " KM/L" << endl;
        cout << "Price: " << price << " Taka" << endl << endl;
    }
};
class Ford: public Car
{
    public:
    string modelType;
    Ford(string modelType, int storeOwnershipCost, int warranty, int
seatingCapacity, string fuelType, double milage, int price):
    Car(storeOwnershipCost, warranty, seatingCapacity, fuelType, milage, price),
modelType(modelType){}
    void showInfo()
    {
        cout << "Model Type: " << modelType << endl;
        cout << "Store Ownership Cost: " << storeOwnershipCost << " Taka" <<
endl;
        cout << "Warranty: " << warranty << " Year" << endl;
        cout << "Seating Capacity: " << seatingCapacity << endl;
        cout << "Fuel Type: " << fuelType << endl;
        cout << "Milage: " << milage << " KM/L" << endl;
        cout << "Price: " << price << " Taka" << endl << endl;
    }
}
```

```
class Walton: public Bike
{
   public:
   string modelType;
};
class TVS: public Bike
{
   public:
   string modelType;
};
int main()
{
   Audi audiCar("Audi R8", 500000, 4, 2, "Petrol", 5.3, 26923000);
   Ford fordCar("Ford Mustang", 200000, 3, 4, "Petrol", 7.9, 6318730);
   audiCar.showInfo();
   fordCar.showInfo();
   return 0;
}
```

23. Create a class named Shape with a function that prints "This is a shape". Create another class named Polygon inheriting the Shape class with the same function that prints "Polygon is a shape". Create two other classes named Rectangle and Triangle having the same function which prints "Rectangle is a polygon" and "Triangle is a polygon" respectively. Again, make another class named Square having the same function which prints "Square is a rectangle". Now, try calling the function by the object of each of these classes.

```
// ©2024, Kabbo Talukder (CSE-56)
#include <bits/stdc++.h>
using namespace std;
```

```
class Shape
{
    public:
    void intro(){
        cout << "This is a shape" << endl;
    }
};

class Polygon: public Shape
{
    public:
    void intro(){
        cout << "Polygon is a shape" << endl;
    }
};

class Rectangle: public Polygon
{
    public:
    void intro(){
        cout << "Rectangle is a polygon" << endl;
    }
};

class Triangle: public Polygon
{
    public:
```

```
    void intro(){
        cout << "Triangle is a polygon" << endl;
    }
};

class Square: public Rectangle
{
    public:
    void intro(){
        cout << "Square is a rectangle" << endl;
    }
};

int main()
{
    Shape shape;
    Polygon polygon;
    Rectangle rectangle;
    Triangle triangle;
    Square square;
    shape.intro();
    polygon.intro();
    rectangle.intro();
    triangle.intro();
    square.intro();
    return 0;
}
```