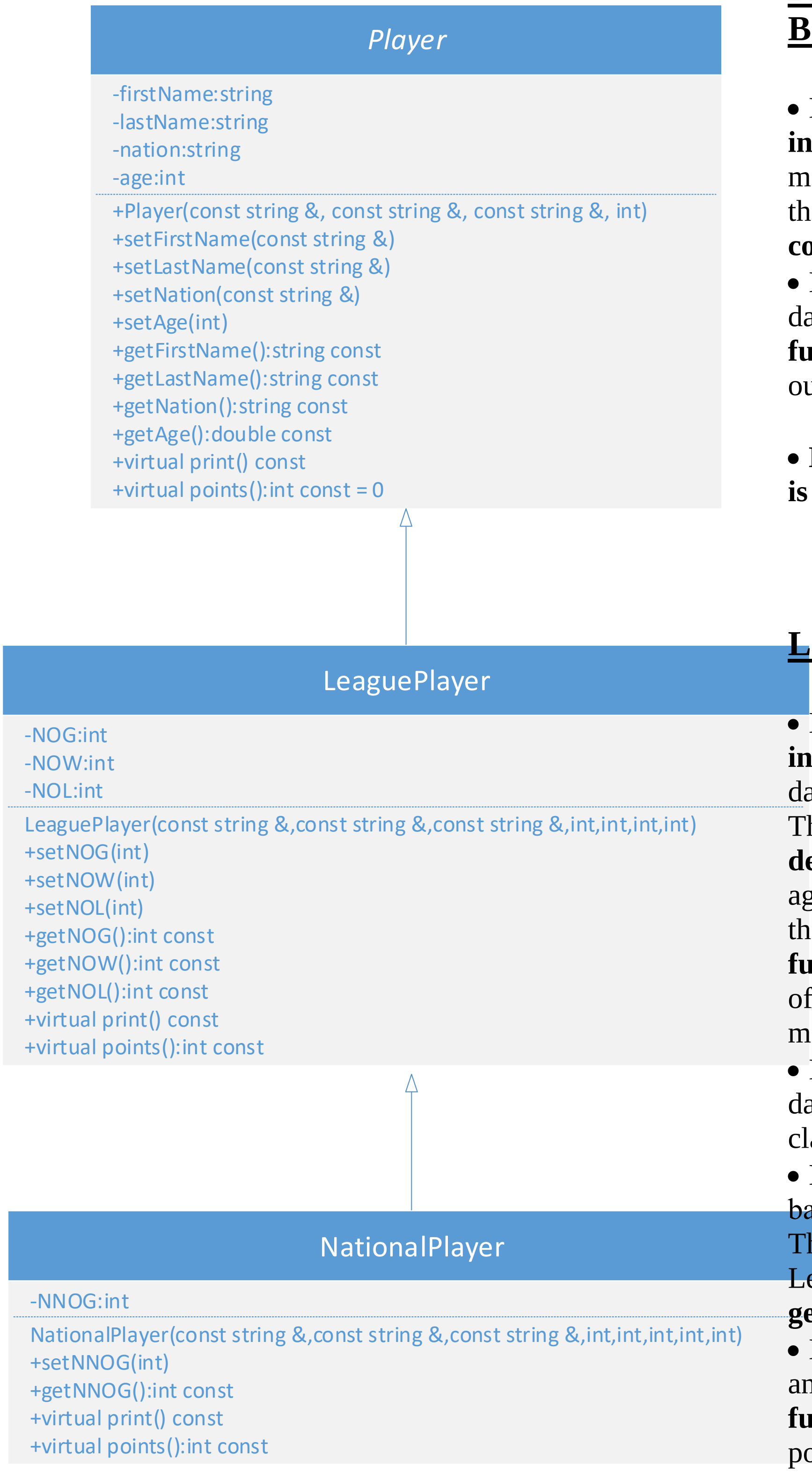
Write a C++ code to implement the UML class diagram.



Player

-firstName:string
-lastName:string
-nation:string
-age:int

+Player(const string &, const string &, const string &, int)
+setFirstName(const string &)
+setLastName(const string &)
+setNation(const string &)
+setAge(int)
+getFirstName():string const
+getLastName():string const
+getNation():string const
+getAge():double const
+virtual print() const
+virtual points():int const = 0

LeaguePlayer

-NOG:int
-NOW:int
-NOL:int

LeaguePlayer(const string &,const string &,const string &,int,int,int,int)
+setNOG(int)
+setNOW(int)
+setNOL(int)
+getNOG():int const
+getNOW():int const
+getNOL():int const
+virtual print() const
+virtual points():int const

NationalPlayer

-NNOG:int

NationalPlayer(const string &,const string &,const string &,int,int,int,int,int)
+setNNOG(int)
+getNNOG():int const
+virtual print() const
+virtual points():int const

Player Class (Abstract Base Class)

- In constructor, use **member initializer** (:) to initialize data members. The constructor of the class is a **default constructor** with 0 for age.
- In print function, print the data members by using **get functions** according to the output, which is given below.

- **Notice that points function is a pure virtual function.**

LeaguePlayer Class

- In constructor, use **base-class initializer syntax** to initialize data members of the base class. The constructor of the class is a **default constructor** with 0 for age, NOG, NOW, and NOL. In the constructor, use **set functions** to set initial values of the LeaguePlayer class data members.
- In set functions, validate the data. Data members of the class must be greater than 0.
- In print function, call the base class's print function. Then print data members of the LeaguePlayer Class by using **get functions**.
- In points function, calculate and return points by using **get functions**. The formula for points is given below:

$$points = 10NOG + 3NOW - 2NOL$$

NationalPlayer Class

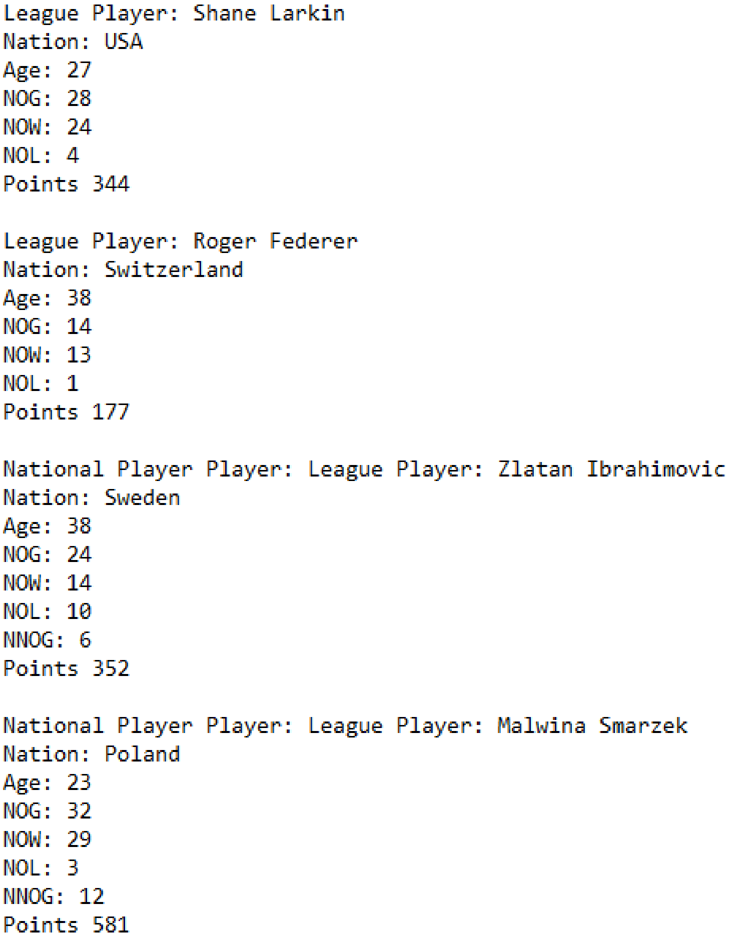
- In constructor, use **base-class initializer syntax** to initialize data members of the base class. The constructor of the class is a **default constructor** with 0 for age, NOG, NOW, NOL, and NNOG. In the constructor, use **set function** to set initial values of the NationalPlayer class data member.
- In set function, validate the data. Data member of the class must be greater than 0.
- In print function, call the base class's print function. Then print data member of the NationalPlayer Class by using **getNNOG function**.
- In points function, calculate and return points. In the function, call base class's points function and **getNNOG function**. The formula for points is given below:

$$points = 10NOG + 3NOW - 2NOL + 15NNOG$$

Driver Program

- Define two objects for LeaguePlayer class and initialize the objects with data given below.
- Define two objects for NationalPlayer class and initialize the objects with data given below.
- Define a **Player *** vector with 4 items. Initialize the vector with the four objects defined for LeaguePlayer and NationalPlayer classes.
- **Use a for loop to call print and points methods**.

The output of the program must be as follows:



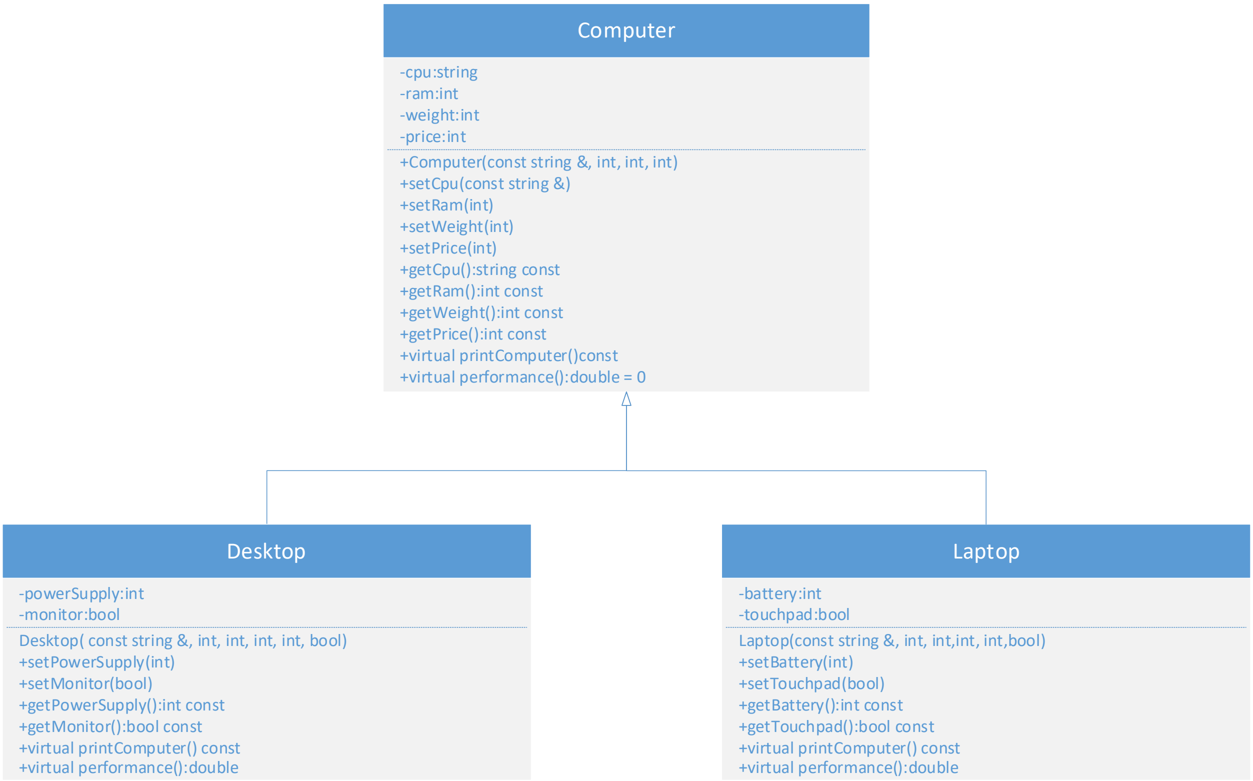
```
League Player: Shane Larkin
Nation: USA
Age: 27
NOG: 28
NOW: 24
NOL: 4
Points 344

League Player: Roger Federer
Nation: Switzerland
Age: 38
NOG: 14
NOW: 13
NOL: 1
Points 177

National Player Player: League Player: Zlatan Ibrahimovic
Nation: Sweden
Age: 38
NOG: 24
NOW: 14
NOL: 10
NNOG: 6
Points 352

National Player Player: League Player: Malwina Smarzek
Nation: Poland
Age: 23
NOG: 32
NOW: 29
NOL: 3
NNOG: 12
Points 581
```

Write a C++ code to implement the UML class diagram.



Computer Class(Abstract Base Class)

- In constructor, use **member initializer** (:) to initialize data members. The constructor of the class is a **default constructor** with 0 for ram, weight, and price.
- In print function, print the data members by using **get functions** according to the output, which is given below.
- **Notice that performance function is a pure virtual function.**

Desktop Class

- The constructor of the class is a **default constructor** for powerSupply with 0 and monitor with false. In constructor, use **base-class initializer syntax** to initialize data members of the base class. Then, use **set functions** to set initial values of the Desktop class data members.
- In set functions, validate the data. Data members of the class must be greater than 0.
- In print function, call base class's print function. Then print data members of the Desktop Class by using **get functions**.
- In performance function, calculate and return performance value by using **get functions**. The formula for performance is given below:

$$performace = 0.3 * \frac{10,000}{price} + 0.7 * \frac{64}{ram}$$

Laptop Class

- The constructor of the class is a **default constructor** for battery with 0 and touchpad with false. In constructor, use **base-class initializer syntax** to initialize data members of the base class. Then, use **set functions** to set initial values of the Laptop class data members.
- In set function, validate the data. Data member of the class must be greater than 0.
- In print function, call base class's print function. Then print data members of the Laptop Class by using **get functions**.

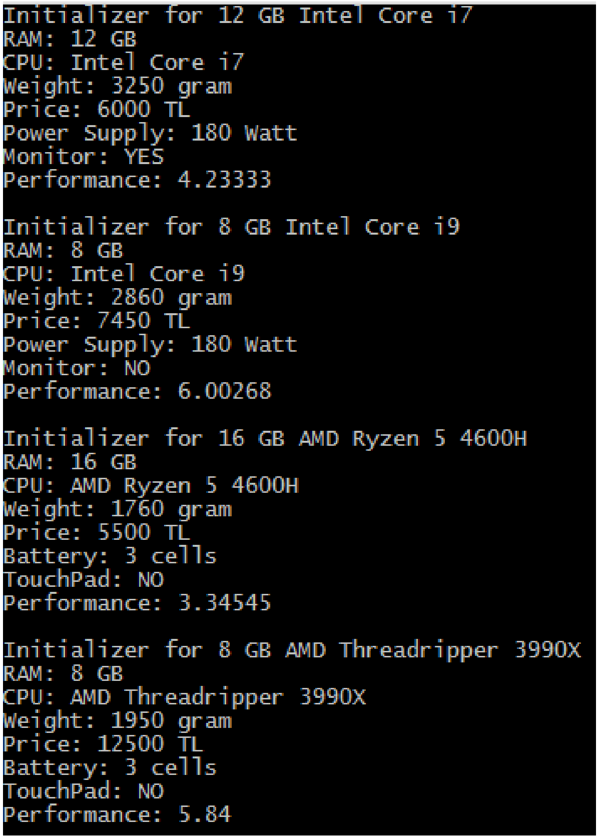
- In performance function, calculate and return performance value by using **get functions**. The formula for performance is given below:

$$performace = 0.3 * \frac{10,000}{price} + 0.7 * \frac{64}{ram}$$

Driver Program

- Define two objects for Desktop class and initialize the objects with data given below.
- Define two objects for Laptop class and initialize the objects with data given below.
- Define a **Computer *** vector with 4 items. Initialize the vector with the four objects defined for Desktop and Laptop classes.
- **Use a for loop to call print and performance method**.

The output of the program must be as follows:



```
Initializer for 12 GB Intel Core i7
RAM: 12 GB
CPU: Intel Core i7
Weight: 3250 gram
Price: 6000 TL
Power Supply: 180 Watt
Monitor: YES
Performance: 4.23333

Initializer for 8 GB Intel Core i9
RAM: 8 GB
CPU: Intel Core i9
Weight: 2860 gram
Price: 7450 TL
Power Supply: 180 Watt
Monitor: NO
Performance: 6.00268

Initializer for 16 GB AMD Ryzen 5 4600H
RAM: 16 GB
CPU: AMD Ryzen 5 4600H
Weight: 1760 gram
Price: 5500 TL
Battery: 3 cells
TouchPad: NO
Performance: 3.34545

Initializer for 8 GB AMD Threadripper 3990X
RAM: 8 GB
CPU: AMD Threadripper 3990X
Weight: 1950 gram
Price: 12500 TL
Battery: 3 cells
TouchPad: NO
Performance: 5.84
```