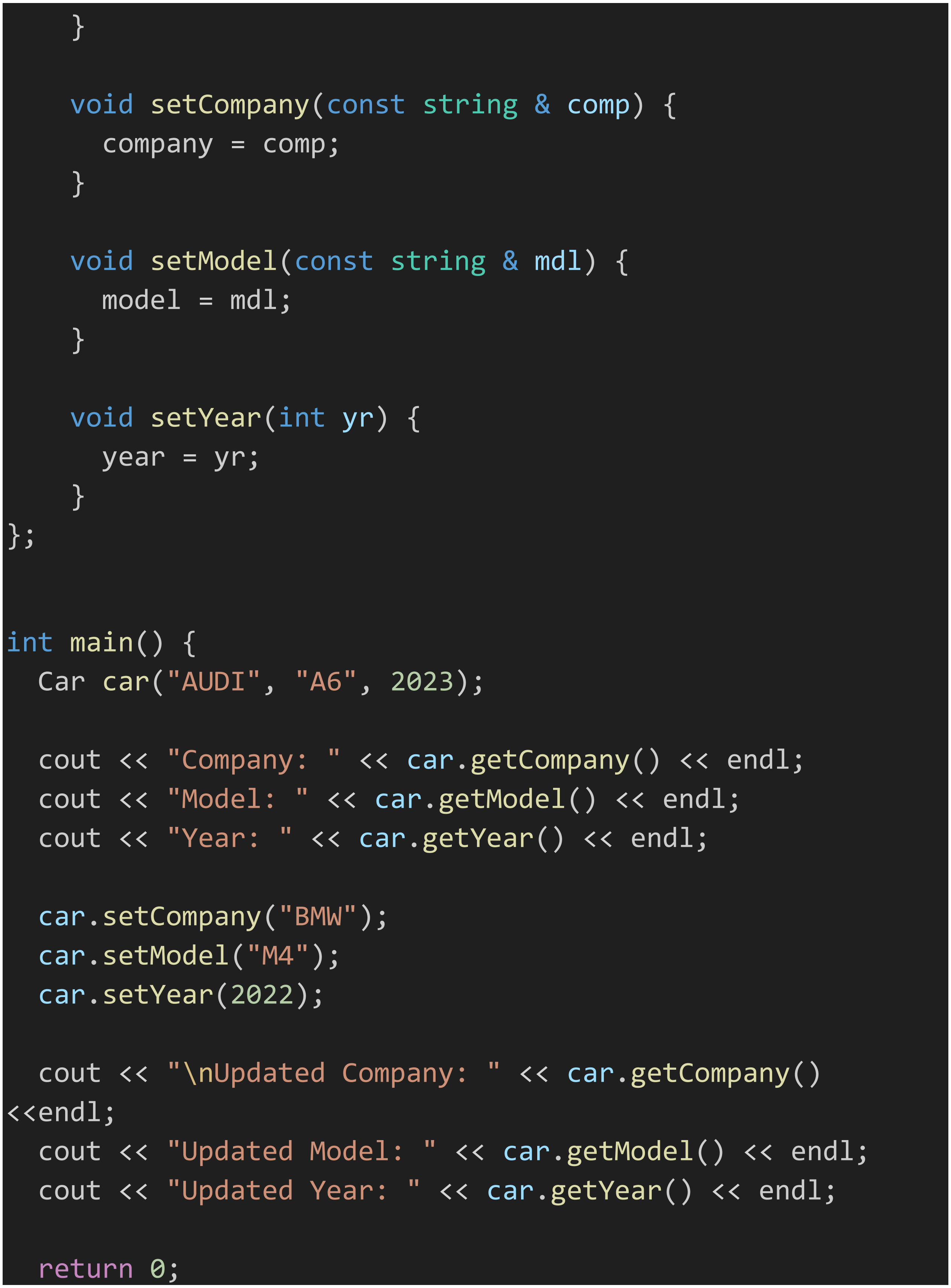
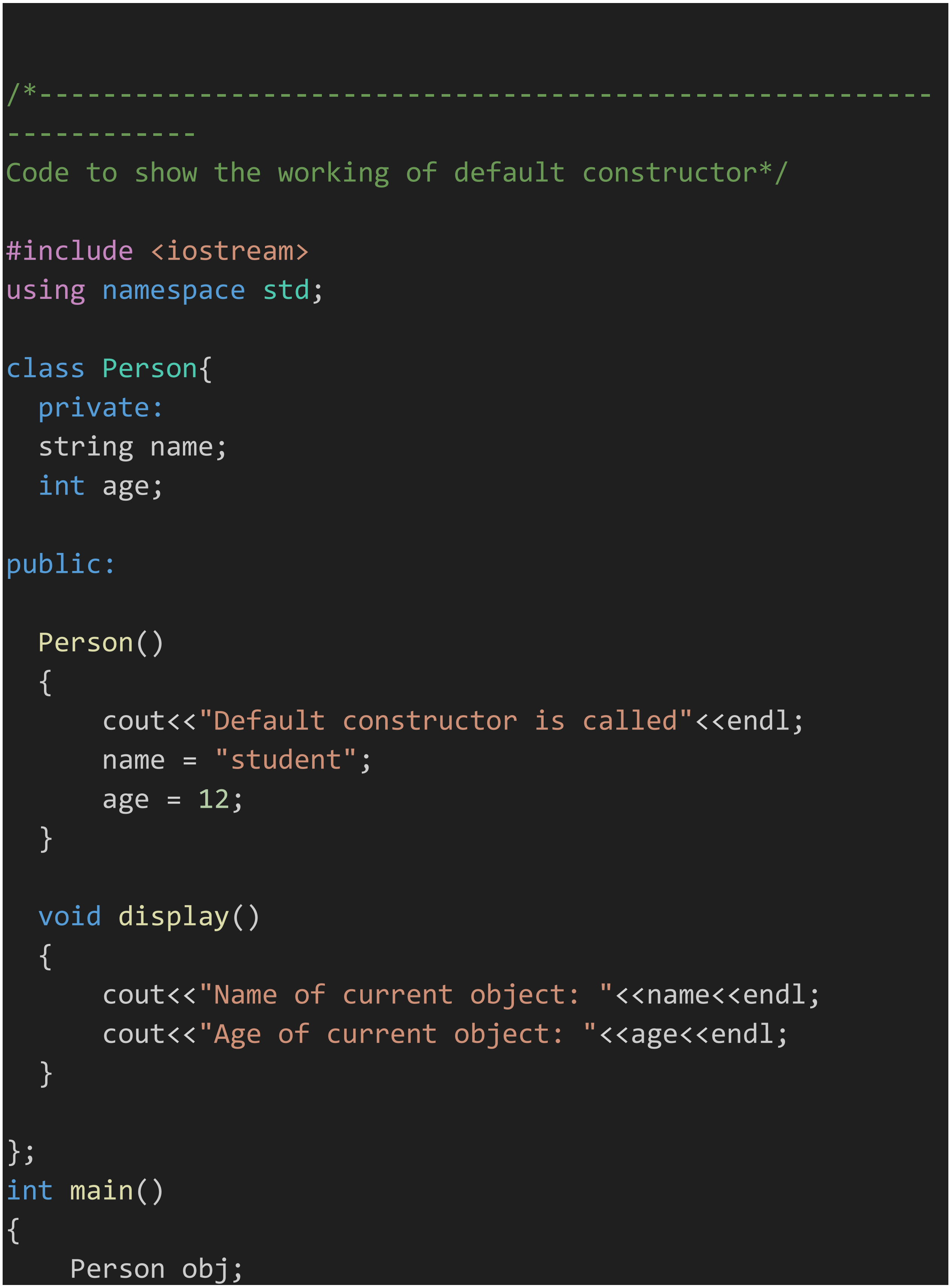
```
/*--------------------------------------------------------
-----------
Write a C++ program to create a class called Car that has
private member variables for company, model, and year.
Implement member functions to get and set these
variables.*/

#include <iostream>
#include <string>

#include <iostream>
#include <string>

class Car {
  private:
    string company;
    string model;
    int year;

  public:
    Car(const string & comp, const string & mdl, int yr):
company(comp), model(mdl), year(yr) {}

    string getCompany() {
      return company;
    }

    string getModel() {
      return model;
    }

    int getYear() {
      return year;
```



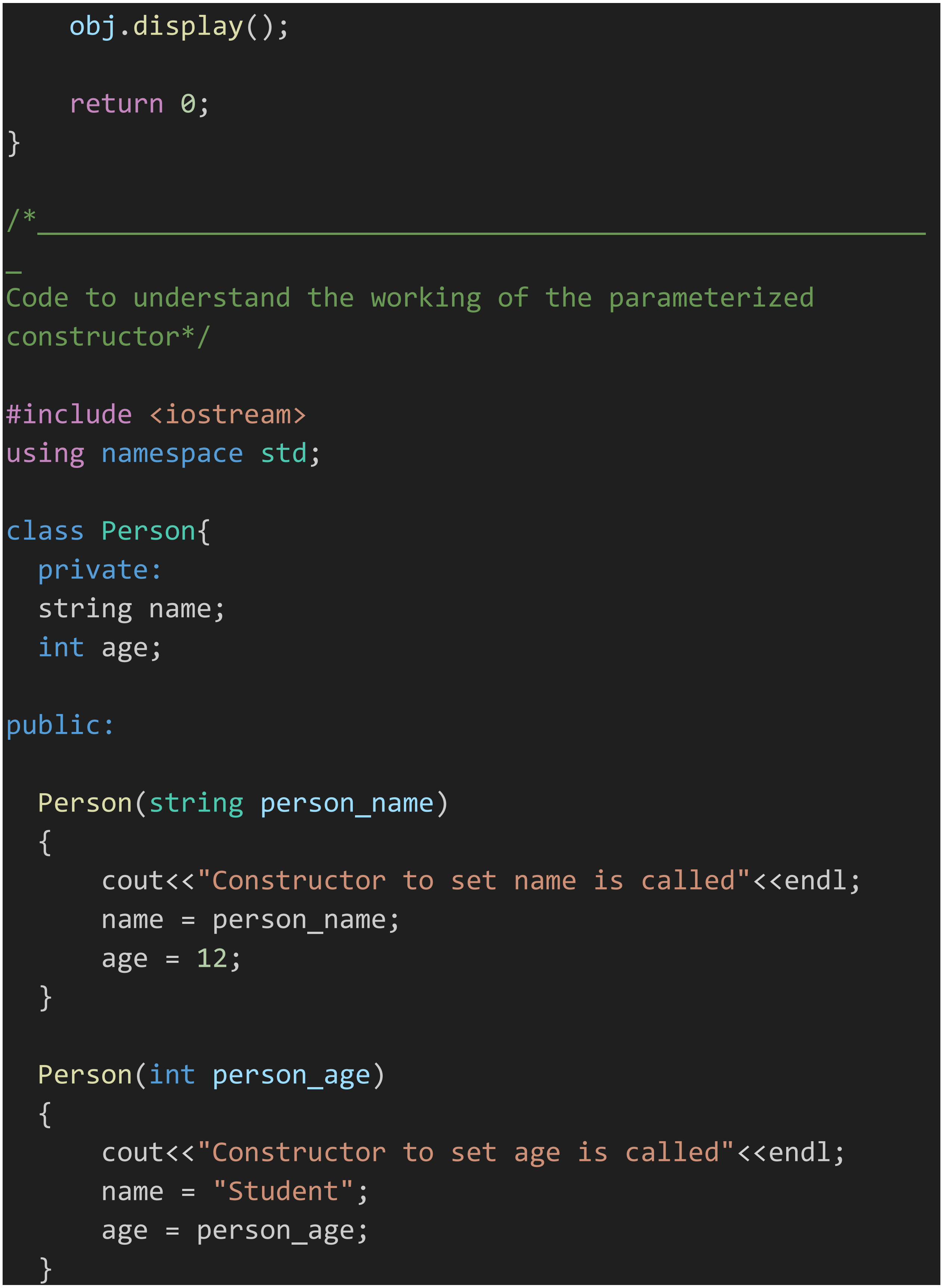
```cpp
    }

    void setCompany(const string & comp) {
      company = comp;
    }

    void setModel(const string & mdl) {
      model = mdl;
    }

    void setYear(int yr) {
      year = yr;
    }
};


int main() {
  Car car("AUDI", "A6", 2023);

  cout << "Company: " << car.getCompany() << endl;
  cout << "Model: " << car.getModel() << endl;
  cout << "Year: " << car.getYear() << endl;

  car.setCompany("BMW");
  car.setModel("M4");
  car.setYear(2022);

  cout << "\nUpdated Company: " << car.getCompany()
<<endl;
  cout << "Updated Model: " << car.getModel() << endl;
  cout << "Updated Year: " << car.getYear() << endl;

  return 0;
```

```
}

/*-----------------------------------------------------------
------------------
Write a C++ program to implement a class called
BankAccount that has private member variables for account
number and balance. Include member functions to deposit
and withdraw money from the account.*/

#include <iostream>
#include <string>

using namespace std;

class BankAccount {
  private:
    string accountNumber;
    double balance;

  public:
    BankAccount(const string & accNum, double
initialBalance): accountNumber(accNum),
balance(initialBalance) {}

    void deposit(double amount) {
      balance += amount;
      cout << "Deposit successful. Current balance: " <<
balance << endl;
    }

    void withdraw(double amount) {
      if (amount <= balance) {
        balance -= amount;
```

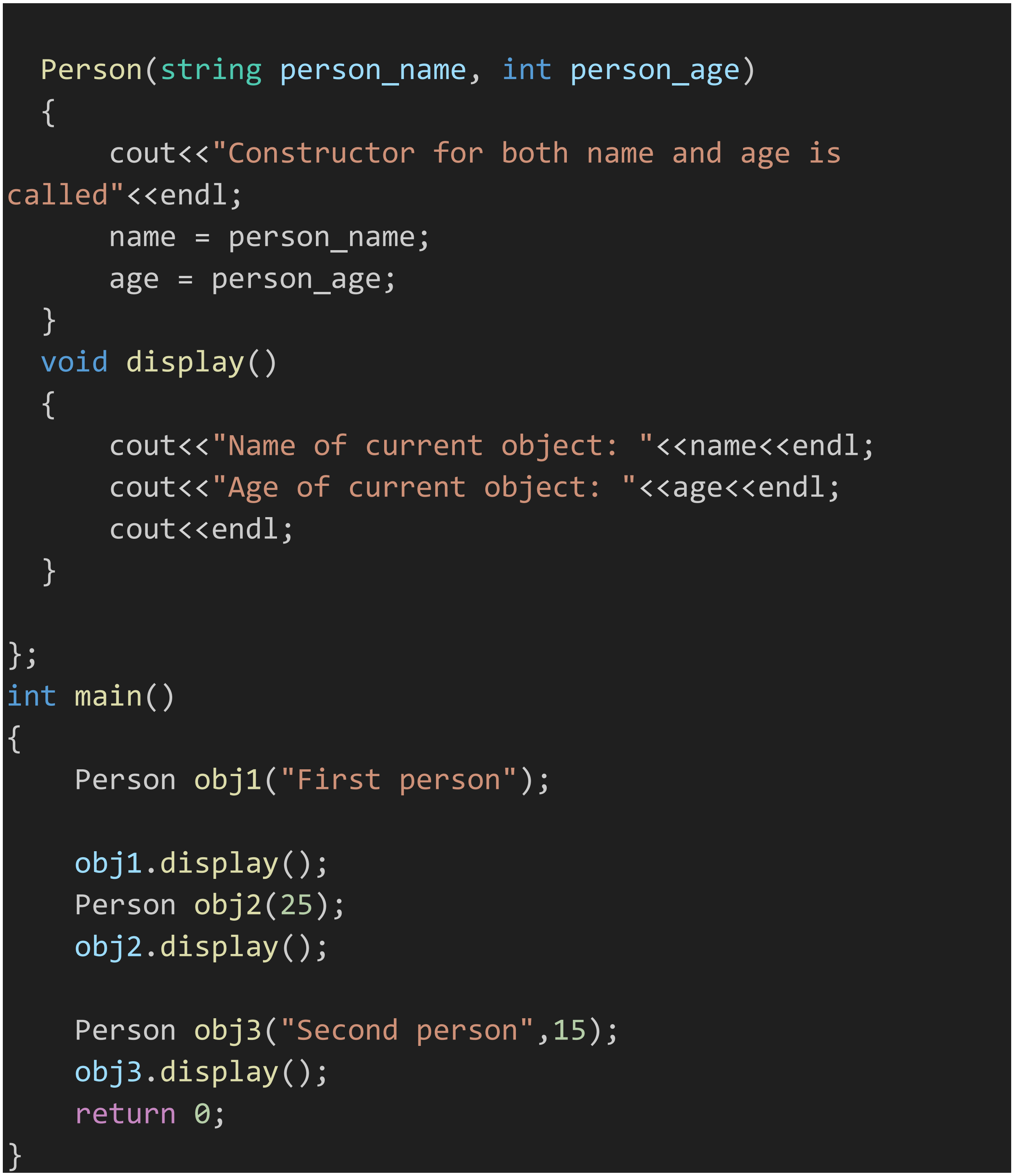
```cpp
        cout << "Withdrawal successful. Current balance: " << balance << endl;
      } else {
        cout << "Insufficient balance. Cannot withdraw." << endl;
      }
    }
};

int main() {
  string sacno = "SB-123";
  double Opening_balance, deposit_amt, withdrawal_amt;
  Opening_balance = 1000;
  cout << "A/c. No." << sacno << " Balance: " << Opening_balance << endl;

  BankAccount account(sacno, 1000.0);

  deposit_amt = 1500;
  cout << "Deposit Amount: " << deposit_amt << endl;
  account.deposit(deposit_amt);

  withdrawal_amt = 750;
  cout << "Withdrawal Amount: " << withdrawal_amt << endl;
  account.withdraw(withdrawal_amt);

  withdrawal_amt = 1800;
  cout << "Attempt to withdrawal Amount: " << withdrawal_amt << endl;
  account.withdraw(withdrawal_amt);

  return 0;
}
```



```
/*--------------------------------------------------------
------------
Code to show the working of default constructor*/

#include <iostream>
using namespace std;

class Person{
  private:
  string name;
  int age;

public:

  Person()
  {
      cout<<"Default constructor is called"<<endl;
      name = "student";
      age = 12;
  }

  void display()
  {
      cout<<"Name of current object: "<<name<<endl;
      cout<<"Age of current object: "<<age<<endl;
  }

};
int main()
{
    Person obj;
```



```
    obj.display();

    return 0;
}

/*________________________________________________________
_
Code to understand the working of the parameterized
constructor*/

#include <iostream>
using namespace std;

class Person{
  private:
  string name;
  int age;

public:

  Person(string person_name)
  {
      cout<<"Constructor to set name is called"<<endl;
      name = person_name;
      age = 12;
  }

  Person(int person_age)
  {
      cout<<"Constructor to set age is called"<<endl;
      name = "Student";
      age = person_age;
  }
```



```
  Person(string person_name, int person_age)
  {
      cout<<"Constructor for both name and age is
called"<<endl;
      name = person_name;
      age = person_age;
  }
  void display()
  {
      cout<<"Name of current object: "<<name<<endl;
      cout<<"Age of current object: "<<age<<endl;
      cout<<endl;
  }

};
int main()
{
    Person obj1("First person");

    obj1.display();
    Person obj2(25);
    obj2.display();

    Person obj3("Second person",15);
    obj3.display();
    return 0;
}
```