```cpp
// ship-final.cpp
/**
 * @author [Mayur M Joshi & Sathkeerthi Y Agnihothri]
 * @create date 2021-07-17 11:05:12
 * @modify date 2021-07-19 13:27:50
 * @desc [Program to maintain a dockyard database]
**/

#include<iostream>
#include<iomanip>
#include <vector>
#include<algorithm>
using namespace std;

void PrintMenu()    //Function to print menu
{
    cout<<endl;
    cout<<"The functions provided and their choices in this database are:\n";
    cout<<"\t1: Enter data of a Ship Docking\n";
    cout<<"\t2: Notification of a Ship Leaving the dock\n";
    cout<<"\t3: Search the details of a Ship\n";
    cout<<"\t4: Get details of all the ships in the dock\n";
    cout<<"\t5: Modify details of a Ship currently at dock\n";
    cout<<"\tOther values: Exit category\n";
    cout<<"Enter your choice:\n";
}
// Base class called Ship
//template <typename X>
class Ship
{
    protected:
            string shipName, captain;
            int yearbuilt;
            //static int count;
    public:
    // Ship() : shipName("no_name"), yearbuilt(0) {count++;}
    Ship(string shipName, string captName, int yearbuilt){
        this -> shipName = shipName;
        this -> captain = captName;
        this -> yearbuilt = yearbuilt;
        //count++;
    }
    virtual void ShipDocking() = 0; //Absolute virtual function
    virtual void ShipLeaving() = 0; //Absolute virtual function
};

//int Ship::count = 0;

// Derived class called CruiseShip
class CruiseShip : public Ship
{
    private:
        int passengerCount; // in numbers
    public:
    CruiseShip():Ship("", "", 0)
    {
        // shipName = "";
        // captain = "";
```

```
59          // yearbuilt = "";
60          passengerCount = 0;
61          //count++;
62      }
63      CruiseShip(string Name, string captName, int year, int passenger):Ship(Name,
captName, year)
64      {
65          // this -> shipName = shipName;
66          // this -> captain = captName;
67          // this -> yearbuilt = yearbuilt;
68          this -> passengerCount = passengerCount;
69          //count++;
70      }
71      void ShipDocking()
72      {
73          cout<<"Enter the details of the Cruise Ship Docking:\n";
74          cin>>shipName>>captain>>yearbuilt>>passengerCount;
75      }
76      void ShipLeaving()
77      {
78          cout<<"The Cruise Ship leaving the dock is:\n";
79          cout<<"Ship Name: "<<shipName<<endl;
80          cout<<"Captain Name: "<<captain<<endl;
81          cout<<"Built Year: "<<yearbuilt<<endl;
82          cout<<"Passenger Count: "<<passengerCount<<endl;
83      }
84      void getdata()
85      {
86          cout<<"Ship Name: "<<shipName<<endl;
87          cout<<"Captain Name: "<<captain<<endl;
88          cout<<"Built Year: "<<yearbuilt<<endl;
89          cout<<"Passenger Count: "<<passengerCount<<endl;
90      }
91      string getCapt()
92      {
93          return captain;
94      }
95  };
96
97  // Derived class called CargoShip
98  template <typename X>
99  class CargoShip : public Ship
100 {
101     private:
102         X cargoCapacity; // in kgs, can be either int or double
103     public:
104     CargoShip():Ship("", "", 0)
105     {
106         cargoCapacity = 0;
107         //count++;
108     }
109     CargoShip(string Name, string captName, int year, X cargoCapacity):Ship(Name,
captName, year)
110     {
111         this -> shipName = shipName;
112         this -> captain = captName;
113         this -> yearbuilt = yearbuilt;
114         this -> cargoCapacity = cargoCapacity;
115         //count++;
```

```
    }
    void ShipDocking()
    {
        cout<<"Enter the details of the Cargo Ship Docking:\n";
        cin>>shipName>>captain>>yearbuilt>>cargoCapacity;
    }
    void ShipLeaving()
    {
        cout<<"The Cargo Ship leaving the dock is:\n";
        cout<<"Ship Name: "<<shipName<<endl;
        cout<<"Captain Name: "<<captain<<endl;
        cout<<"Built Year: "<<yearbuilt<<endl;
        cout<<"Cargo Capacity: "<<cargoCapacity<<endl;
    }
    void getdata()
    {
        cout<<"Ship Name: "<<shipName<<endl;
        cout<<"Captain Name: "<<captain<<endl;
        cout<<"Built Year: "<<yearbuilt<<endl;
        cout<<"Cargo Capacity: "<<cargoCapacity<<endl;
    }
    string getCapt()
    {
        return captain;
    }
};

// Derived class called WarShip
template <typename X>
class WarShip : public Ship{
    X weaponCapacity; // in kgs, can be either int or double
public:
    // Warship(string Name = "", string captain = "", int year = 0, int cap =
0):Ship(Name, captain, year)
    // {
    //     weaponCapacity = cap;
    //     //count++;
    // }
    WarShip():Ship("", "", 0)
    {
        // shipName = "";
        // captain = "";
        // yearbuilt = 0;
        weaponCapacity = 0;
    }
    WarShip(string shipName, string captName, int yearbuilt, X weaponCapacity)
    {
        this -> shipName = shipName;
        this -> captain = captName;
        this -> yearbuilt = yearbuilt;
        this -> weaponCapacity = weaponCapacity;
        //count++;
    }
    void ShipDocking()
    {
        cout<<"Enter the details of the War Ship Docking:\n";
        cin>>shipName>>captain>>yearbuilt>>weaponCapacity;
    }
    void ShipLeaving()
```

```
    {
        cout<<"The War Ship leaving the dock is:\n";
        cout<<"Ship Name: "<<shipName<<endl;
        cout<<"Captain Name: "<<captain<<endl;
        cout<<"Built Year: "<<yearbuilt<<endl;
        cout<<"Weapon Capacity: "<<weaponCapacity<<endl;
    }
    void getdata()
    {
        cout<<"Ship Name: "<<shipName<<endl;
        cout<<"Captain Name: "<<captain<<endl;
        cout<<"Built Year: "<<yearbuilt<<endl;
        cout<<"Weapon Capacity: "<<weaponCapacity<<endl;
    }
    string getCapt()
    {
        return captain;
    }
};

int main()
{
    CruiseShip c;
    CargoShip<int> car;
    WarShip<int> w;
    vector<CruiseShip> cruise;
    vector< CargoShip<int> > cargo;
    vector< WarShip<int> > war;
    int choice, type, i;
    string ch, capt;
    vector<CruiseShip>::iterator cr;
    vector<CargoShip<int> >::iterator ca;
    vector<WarShip<int> >::iterator wa;
    cout<<"\t\tWelcome to RMS Docks:\n";
    do
    {
        cout<<"Enter the Type of Ship to be operated on:\n";
        cout<<"Enter 1 for CruiseShip, 2 for CargoShip, 3 for WarShip:\n";
        cin>>type;
        switch (type)
        {
// → CruiseShip data
        case 1:
            cout<<"You have Selected CruiseShip type:\n";
            do
            {
                PrintMenu();
                cin>>choice;
                switch (choice)
                {
                case 1:
                    c.ShipDocking();
                    cruise.push_back(c);
                    cout << endl;
                    break;
                case 2:
                    if(cruise.empty())
                    {
                        cout<<"Error! No CruiseShips in dock.\n";
```

```
                    break;
                }
                cout<<"Enter the captain of the ship leaving the dock:\n";
                cin>>capt;
                for(cr = cruise.begin();cr<=cruise.end();cr++)
                {
                    if(cr->getCapt() == capt)
                    {
                        cr->ShipLeaving();
                        cruise.erase(cr);
                        break;
                    }
                    else if(cr == cruise.end())
                    {
                        cout<<"Error! No captain found\n";
                        break;
                    }
                }
                // cout << endl;
                break;
            case 3:
                if(cruise.empty())
                {
                    cout<<"Error! No CruiseShips in dock.\n";
                    break;
                }
                cout<<"Enter the captain of the ship whose details are
required:\n";
                cin>>capt;
                //vector<CruiseShip>::iterator it;
                for(cr = cruise.begin();cr<=cruise.end();cr++)
                {
                    if(cr->getCapt() == capt)
                    {
                        cout<<"The details requested are:\n";
                        cr->getdata();
                        break;
                    }
                    else if(cr == cruise.end())
                    {
                        cout<<"Error! No captain found\n";
                        break;
                    }
                }
                // cout << endl;
                break;
            case 4:
                int i;
                if(cruise.empty())
                {
                    cout<<"The dock has no Cruiseships currently.\n";
                    break;
                }
                cout<<"The details of all the ships are:\n";
                for(cr = cruise.begin(), i=1;cr<cruise.end();cr++, i++)
                {
                    cout<<"The details of Ship "<<i<<":"<<endl;
                    cr->getdata();
                    cout<<endl;
```

```
                    }
                    // cout << endl;
                    break;
                case 5:
                    if(cruise.empty())
                    {
                        cout<<"Error! No CruiseShips in dock.\n";
                        break;
                    }
                    cout<<"Enter the captain of the ship whose details are to be changed:\n";
                    cin>>capt;
                    for(cr = cruise.begin();cr<=cruise.end();cr++)
                    {
                        if(cr->getCapt() == capt)
                        {
                            cout<<"The Ship details will be changed:\n";
                            cr->ShipDocking();
                            break;
                        }
                        else if(cr == cruise.end())
                        {
                            cout<<"Error! No captain found\n";
                            break;
                        }
                    }
                    // cout << endl;
                    break;
                }
            }
            while(choice <= 5);
            cout<<"Exiting category.\n" << endl;
            break;
// -> CargoShip data
        case 2:
            cout<<"You have Selected CargoShip type:\n";
            //CargoShip<int> car;
            do
            {
                PrintMenu();
                cin>>choice;
                switch (choice)
                {
                case 1:
                    car.ShipDocking();
                    cargo.push_back(car);
                    cout << endl;
                    break;
                case 2:
                    if(cargo.empty())
                    {
                        cout<<"Error! No Cargoships in dock.\n";
                        break;
                    }
                    cout<<"Enter the captain of the ship leaving the dock:\n";
                    cin>>capt;
                    for(ca = cargo.begin();ca<=cargo.end();ca++)
                    {
                        if(ca->getCapt() == capt)
```

```
                    {
                        ca→ShipLeaving();
                        cargo.erase(ca);
                        break;
                    }
                    else if(ca == cargo.end())
                    {
                        cout<<"Error! No captain found\n";
                        break;
                    }
                }
                //cout << endl;
                break;
            case 3:
                if(cargo.empty())
                {
                    cout<<"Error! No Cargoships in dock.\n";
                    break;
                }
                cout<<"Enter the captain of the ship whose details are
required:\n";
                cin>>capt;
                //vector<CargoShip>::iterator it;
                for(ca = cargo.begin();ca⩽cargo.end();ca++)
                {
                    if(ca→getCapt() == capt)
                    {
                        cout<<"The details requested are:\n";
                        ca→getdata();
                        break;
                    }
                    else if(ca == cargo.end())
                    {
                        cout<<"Error! No captain found\n";
                        break;
                    }
                }
                //cout << endl;
                break;
            case 4:
                if(cargo.empty())
                {
                    cout<<"The dock has no Cargoships currently.\n";
                    break;
                }
                cout<<"The details of all the ships are:\n";
                for(ca = cargo.begin(), i=1;ca<cargo.end();ca++, i++)
                {
                    cout<<"The details of Ship "<<i<<":"<<endl;
                    ca→getdata();
                    cout<<endl;
                }
                //cout << endl;
                break;
            case 5:
                if(cargo.empty())
                {
                    cout<<"Error! No Cargoships in dock.\n";
                    break;
```

```
                    }
                    cout<<"Enter the captain of the ship whose details are to be changed:\n";
                    cin>>capt;
                    for(ca = cargo.begin();ca<=cargo.end();ca++)
                    {
                        if(ca->getCapt() == capt)
                        {
                            cout<<"The Ship details will be changed:\n";
                            ca->ShipDocking();
                            break;
                        }
                        else if(ca == cargo.end())
                        {
                            cout<<"Error! No captain found\n";
                            break;
                        }
                    }
                    //cout << endl;
                    break;
                }
            }
            while(choice <= 5);
            cout<<"Exiting category.\n" << endl;
            break;
// -> WarShip data
        case 3:
            cout<<"You have Selected WarShip type:\n";
            //Warship<int> w;
            do
            {
                PrintMenu();
                cin>>choice;
                switch (choice)
                {
                case 1:
                    w.ShipDocking();
                    war.push_back(w);
                    cout << endl;
                    break;
                case 2:
                    if(war.empty())
                    {
                        cout<<"Error! No Warships in dock.\n";
                        break;
                    }
                    cout<<"Enter the captain of the ship leaving the dock:\n";
                    cin>>capt;
                    for(wa = war.begin();wa<=war.end();wa++)
                    {
                        if(wa->getCapt() == capt)
                        {
                            wa->ShipLeaving();
                            war.erase(wa);
                            break;
                        }
                        else if(wa == war.end())
                        {
                            cout<<"Error! No captain found\n";
```

```
                        break;
                    }
                }
                //cout << endl;
                break;
            case 3:
                if(war.empty())
                {
                    cout<<"Error! No Warships in dock.\n";
                    break;
                }
                cout<<"Enter the captain of the ship whose details are
required:\n";
                cin>>capt;
                for(wa = war.begin();wa<=war.end();wa++)
                {
                    if(wa->getCapt() == capt)
                    {
                        cout<<"The details requested are:\n";
                        wa->getdata();
                        break;
                    }
                    else if(wa == war.end())
                    {
                        cout<<"Error! No captain found\n";
                        break;
                    }
                }
                //cout << endl;
                break;
            case 4:
                if(war.empty())
                {
                    cout<<"The dock has no Warships currently.\n";
                    break;
                }
                cout<<"The details of all the ships are:\n";
                for(wa = war.begin(), i=1;wa<war.end();wa++, i++)
                {
                    cout<<"The details of Ship "<<i<<":"<<endl;
                    wa->getdata();
                    cout<<endl;
                }
                //cout << endl;
                break;
            case 5:
                if(war.empty())
                {
                    cout<<"Error! No Warships in dock.\n";
                    break;
                }
                cout<<"Enter the captain of the ship whose details are to be
changed:\n";
                cin>>capt;
                for(wa = war.begin();wa<=war.end();wa++)
                {
                    if(wa->getCapt() == capt)
                    {
                        cout<<"The Ship details will be changed:\n";
```

```
                        wa→ShipDocking();
                        break;
                    }
                    else if(wa == war.end())
                    {
                        cout<<"Error! No captain found\n";
                        break;
                    }
                }
                //cout << endl;
                break;
            }
        }
        while(choice ⩽ 5);
        cout<<"Exiting category.\n" << endl;
        break;
    default:
        cout<<"Error! Invalid input\n" << endl;
        break;
    }
    cout<<"Do you want to continue?(Yes/No):\n";
    cin>>ch;
}while(ch ≠ "No");
return 0;
}
```