

```

1 // ship-final.cpp
2 /**
3  * @author [Mayur M Joshi & Sathkeerthi Y Agnihothri]
4  * @create date 2021-07-17 11:05:12
5  * @modify date 2021-07-19 13:27:50
6  * @desc [Program to maintain a dockyard database]
7  */
8
9 #include<iostream>
10 #include<iomanip>
11 #include <vector>
12 #include<algorithm>
13 using namespace std;
14
15 void PrintMenu()    //Function to print menu
16 {
17     cout<<endl;
18     cout<<"The functions provided and their choices in this database are:\n";
19     cout<<"\t1: Enter data of a Ship Docking\n";
20     cout<<"\t2: Notification of a Ship Leaving the dock\n";
21     cout<<"\t3: Search the details of a Ship\n";
22     cout<<"\t4: Get details of all the ships in the dock\n";
23     cout<<"\t5: Modify details of a Ship currently at dock\n";
24     cout<<"\t0: Other values: Exit category\n";
25     cout<<"Enter your choice:\n";
26 }
27 // Base class called Ship
28 //template <typename X>
29 class Ship
30 {
31     protected:
32         string shipName, captain;
33         int yearbuilt;
34         //static int count;
35     public:
36         // Ship() : shipName("no_name"), yearbuilt(0) {count++;}
37         Ship(string shipName, string captName, int yearbuilt){
38             this → shipName = shipName;
39             this → captain = captName;
40             this → yearbuilt = yearbuilt;
41             //count++;
42         }
43         virtual void ShipDocking() = 0; //Absolute virtual function
44         virtual void ShipLeaving() = 0; //Absolute virtual function
45 };
46
47 //int Ship::count = 0;
48
49 // Derived class called CruiseShip
50 class CruiseShip : public Ship
51 {
52     private:
53         int passengerCount; // in numbers
54     public:
55         CruiseShip():Ship("", "", 0)
56         {
57             // shipName = "";
58             // 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++;

```

```

116 }
117 void ShipDocking()
118 {
119     cout<<"Enter the details of the Cargo Ship Docking:\n";
120     cin>>shipName>>captain>>yearbuilt>>cargoCapacity;
121 }
122 void ShipLeaving()
123 {
124     cout<<"The Cargo Ship leaving the dock is:\n";
125     cout<<"Ship Name: "<<shipName<<endl;
126     cout<<"Captain Name: "<<captain<<endl;
127     cout<<"Built Year: "<<yearbuilt<<endl;
128     cout<<"Cargo Capacity: "<<cargoCapacity<<endl;
129 }
130 void getdata()
131 {
132     cout<<"Ship Name: "<<shipName<<endl;
133     cout<<"Captain Name: "<<captain<<endl;
134     cout<<"Built Year: "<<yearbuilt<<endl;
135     cout<<"Cargo Capacity: "<<cargoCapacity<<endl;
136 }
137 string getCapt()
138 {
139     return captain;
140 }
141 };
142
143 // Derived class called WarShip
144 template <typename X>
145 class WarShip : public Ship{
146     X weaponCapacity; // in kgs, can be either int or double
147 public:
148     // Warship(string Name = "", string captain = "", int year = 0, int cap =
0):Ship(Name, captain, year)
149     // {
150     //     weaponCapacity = cap;
151     //     //count++;
152     // }
153     WarShip():Ship("", "", 0)
154     {
155         // shipName = "";
156         // captain = "";
157         // yearbuilt = 0;
158         weaponCapacity = 0;
159     }
160     WarShip(string shipName, string captName, int yearbuilt, X weaponCapacity)
161     {
162         this → shipName = shipName;
163         this → captain = captName;
164         this → yearbuilt = yearbuilt;
165         this → weaponCapacity = weaponCapacity;
166         //count++;
167     }
168     void ShipDocking()
169     {
170         cout<<"Enter the details of the War Ship Docking:\n";
171         cin>>shipName>>captain>>yearbuilt>>weaponCapacity;
172     }
173     void ShipLeaving()

```

```

174     {
175         cout<<"The War Ship leaving the dock is:\n";
176         cout<<"Ship Name: "<<shipName<<endl;
177         cout<<"Captain Name: "<<captain<<endl;
178         cout<<"Built Year: "<<yearbuilt<<endl;
179         cout<<"Weapon Capacity: "<<weaponCapacity<<endl;
180     }
181 void getdata()
182 {
183     cout<<"Ship Name: "<<shipName<<endl;
184     cout<<"Captain Name: "<<captain<<endl;
185     cout<<"Built Year: "<<yearbuilt<<endl;
186     cout<<"Weapon Capacity: "<<weaponCapacity<<endl;
187 }
188 string getCapt()
189 {
190     return captain;
191 }
192 };
193
194 int main()
195 {
196     CruiseShip c;
197     CargoShip<int> car;
198     WarShip<int> w;
199     vector<CruiseShip> cruise;
200     vector< CargoShip<int> > cargo;
201     vector< WarShip<int> > war;
202     int choice, type, i;
203     string ch, capt;
204     vector<CruiseShip>::iterator cr;
205     vector<CargoShip<int> >::iterator ca;
206     vector<WarShip<int> >::iterator wa;
207     cout<<"\t\tWelcome to RMS Docks:\n";
208     do
209     {
210         cout<<"Enter the Type of Ship to be operated on:\n";
211         cout<<"Enter 1 for CruiseShip, 2 for CargoShip, 3 for WarShip:\n";
212         cin>>type;
213         switch (type)
214         {
215             // → CruiseShip data
216             case 1:
217                 cout<<"You have Selected CruiseShip type:\n";
218                 do
219                 {
220                     PrintMenu();
221                     cin>>choice;
222                     switch (choice)
223                     {
224                         case 1:
225                             c.ShipDocking();
226                             cruise.push_back(c);
227                             cout << endl;
228                             break;
229                         case 2:
230                             if(cruise.empty())
231                             {
232                                 cout<<"Error! No CruiseShips in dock.\n";

```

```

233         break;
234     }
235     cout<<"Enter the captain of the ship leaving the dock:\n";
236     cin>>capt;
237     for(cr = cruise.begin();cr<=cruise.end();cr++)
238     {
239         if(cr->getCapt() == capt)
240         {
241             cr->ShipLeaving();
242             cruise.erase(cr);
243             break;
244         }
245         else if(cr == cruise.end())
246         {
247             cout<<"Error! No captain found\n";
248             break;
249         }
250     }
251     // cout << endl;
252     break;
253 case 3:
254     if(cruise.empty())
255     {
256         cout<<"Error! No CruiseShips in dock.\n";
257         break;
258     }
259     cout<<"Enter the captain of the ship whose details are
required:\n";
260     cin>>capt;
261     //vector<CruiseShip>::iterator it;
262     for(cr = cruise.begin();cr<=cruise.end();cr++)
263     {
264         if(cr->getCapt() == capt)
265         {
266             cout<<"The details requested are:\n";
267             cr->getdata();
268             break;
269         }
270         else if(cr == cruise.end())
271         {
272             cout<<"Error! No captain found\n";
273             break;
274         }
275     }
276     // cout << endl;
277     break;
278 case 4:
279     int i;
280     if(cruise.empty())
281     {
282         cout<<"The dock has no Cruiseships currently.\n";
283         break;
284     }
285     cout<<"The details of all the ships are:\n";
286     for(cr = cruise.begin(), i=1;cr<cruise.end();cr++, i++)
287     {
288         cout<<"The details of Ship "<<i<<":"<<endl;
289         cr->getdata();
290         cout<<endl;

```

```

291     }
292     // cout << endl;
293     break;
294 case 5:
295     if(cruise.empty())
296     {
297         cout<<"Error! No CruiseShips in dock.\n";
298         break;
299     }
300     cout<<"Enter the captain of the ship whose details are to be
changed:\n";
301     cin>>capt;
302     for(cr = cruise.begin();cr<=cruise.end();cr++)
303     {
304         if(cr->getCapt() == capt)
305         {
306             cout<<"The Ship details will be changed:\n";
307             cr->ShipDocking();
308             break;
309         }
310         else if(cr == cruise.end())
311         {
312             cout<<"Error! No captain found\n";
313             break;
314         }
315     }
316     // cout << endl;
317     break;
318 }
319 }
320 while(choice <= 5);
321 cout<<"Exiting category.\n" << endl;
322 break;
323 // → CargoShip data
324 case 2:
325     cout<<"You have Selected CargoShip type:\n";
326     //CargoShip<int> car;
327     do
328     {
329         PrintMenu();
330         cin>>choice;
331         switch (choice)
332         {
333             case 1:
334                 car.ShipDocking();
335                 cargo.push_back(car);
336                 cout << endl;
337                 break;
338             case 2:
339                 if(cargo.empty())
340                 {
341                     cout<<"Error! No Cargoships in dock.\n";
342                     break;
343                 }
344                 cout<<"Enter the captain of the ship leaving the dock:\n";
345                 cin>>capt;
346                 for(ca = cargo.begin();ca<=cargo.end();ca++)
347                 {
348                     if(ca->getCapt() == capt)

```

```

349         {
350             ca→ShipLeaving();
351             cargo.erase(ca);
352             break;
353         }
354     else if(ca == cargo.end())
355     {
356         cout<<"Error! No captain found\n";
357         break;
358     }
359 }
360 //cout << endl;
361 break;
362 case 3:
363     if(cargo.empty())
364     {
365         cout<<"Error! No Cargoships in dock.\n";
366         break;
367     }
368     cout<<"Enter the captain of the ship whose details are
required:\n";
369     cin>>capt;
370     //vector<CargoShip>::iterator it;
371     for(ca = cargo.begin();ca≤cargo.end();ca++)
372     {
373         if(ca→getCapt() == capt)
374         {
375             cout<<"The details requested are:\n";
376             ca→getdata();
377             break;
378         }
379         else if(ca == cargo.end())
380         {
381             cout<<"Error! No captain found\n";
382             break;
383         }
384     }
385     //cout << endl;
386     break;
387 case 4:
388     if(cargo.empty())
389     {
390         cout<<"The dock has no Cargoships currently.\n";
391         break;
392     }
393     cout<<"The details of all the ships are:\n";
394     for(ca = cargo.begin(), i=1;ca<cargo.end();ca++, i++)
395     {
396         cout<<"The details of Ship "<<i<<":"<<endl;
397         ca→getdata();
398         cout<<endl;
399     }
400     //cout << endl;
401     break;
402 case 5:
403     if(cargo.empty())
404     {
405         cout<<"Error! No Cargoships in dock.\n";
406         break;

```

```

407     }
408     cout<<"Enter the captain of the ship whose details are to be
changed:\n";
409     cin>>capt;
410     for(ca = cargo.begin();ca<= cargo.end();ca++)
411     {
412         if(ca->getCapt() == capt)
413         {
414             cout<<"The Ship details will be changed:\n";
415             ca->ShipDocking();
416             break;
417         }
418         else if(ca == cargo.end())
419         {
420             cout<<"Error! No captain found\n";
421             break;
422         }
423     }
424     //cout << endl;
425     break;
426 }
427 }
428 while(choice <= 5);
429 cout<<"Exiting category.\n" << endl;
430 break;
431 // → WarShip data
432 case 3:
433     cout<<"You have Selected WarShip type:\n";
434     //Warship<int> w;
435     do
436     {
437         PrintMenu();
438         cin>>choice;
439         switch (choice)
440         {
441             case 1:
442                 w.ShipDocking();
443                 war.push_back(w);
444                 cout << endl;
445                 break;
446             case 2:
447                 if(war.empty())
448                 {
449                     cout<<"Error! No Warships in dock.\n";
450                     break;
451                 }
452                 cout<<"Enter the captain of the ship leaving the dock:\n";
453                 cin>>capt;
454                 for(wa = war.begin();wa<= war.end();wa++)
455                 {
456                     if(wa->getCapt() == capt)
457                     {
458                         wa->ShipLeaving();
459                         war.erase(wa);
460                         break;
461                     }
462                     else if(wa == war.end())
463                     {
464                         cout<<"Error! No captain found\n";

```



```

465         break;
466     }
467 }
468 //cout << endl;
469 break;
470 case 3:
471     if(war.empty())
472     {
473         cout<<"Error! No Warships in dock.\n";
474         break;
475     }
476     cout<<"Enter the captain of the ship whose details are
required:\n";
477     cin>>capt;
478     for(wa = war.begin();wa<=war.end();wa++)
479     {
480         if(wa->getCapt() == capt)
481         {
482             cout<<"The details requested are:\n";
483             wa->getdata();
484             break;
485         }
486         else if(wa == war.end())
487         {
488             cout<<"Error! No captain found\n";
489             break;
490         }
491     }
492     //cout << endl;
493     break;
494 case 4:
495     if(war.empty())
496     {
497         cout<<"The dock has no Warships currently.\n";
498         break;
499     }
500     cout<<"The details of all the ships are:\n";
501     for(wa = war.begin(), i=1;wa<war.end();wa++, i++)
502     {
503         cout<<"The details of Ship " <<i<<": " <<endl;
504         wa->getdata();
505         cout<<endl;
506     }
507     //cout << endl;
508     break;
509 case 5:
510     if(war.empty())
511     {
512         cout<<"Error! No Warships in dock.\n";
513         break;
514     }
515     cout<<"Enter the captain of the ship whose details are to be
changed:\n";
516     cin>>capt;
517     for(wa = war.begin();wa<=war.end();wa++)
518     {
519         if(wa->getCapt() == capt)
520         {
521             cout<<"The Ship details will be changed:\n";

```

```
522         wa→ShipDocking();
523         break;
524     }
525     else if(wa == war.end())
526     {
527         cout<<"Error! No captain found\n";
528         break;
529     }
530 }
531 //cout << endl;
532 break;
533 }
534 }
535 while(choice ≤ 5);
536 cout<<"Exiting category.\n" << endl;
537 break;
538 default:
539     cout<<"Error! Invalid input\n" << endl;
540     break;
541 }
542 cout<<"Do you want to continue?(Yes/No):\n";
543 cin>>ch;
544 }while(ch ≠ "No");
545 return 0;
546 }
```