

```
1 #include <iostream>
2 #include <iomanip>
3 #include <string>
4 using namespace std;
5 class Laptop
6 {
7 private:
8     string manufacturer;
9     string processor;
10    double screenSize;
11    string color;
12    bool isNew;
13 public:
14    string getManufacturer() const;
15    string getProcessor() const;
16    double getScreenSize() const;
17    string getColor() const;
18    bool getIsNew() const;
19    void setManufacturer(string);
20    void setProcessor(string);
21    void setScreenSize(double);
22    void setColor(string);
23    void setIsNew(bool);
24    void input();
25    void output() const;
26    Laptop();
27    Laptop(string, string = "AMD Ryzen", double = 17.3, string = "Silver",
           bool = false);
28    bool friend operator==(const Laptop& a, const Laptop& b);
29 };
30 Laptop::Laptop()
31 {
32     manufacturer = "Asus";
33     processor = "Intel i5";
34     screenSize = 15.6;
35     color = "Black";
36     isNew = true;
37 }
38 Laptop::Laptop(string a, string b, double c, string d, bool e)
39 {
40     manufacturer = a;
41     processor = b;
42     screenSize = c;
43     color = d;
44     isNew = e;
45 }
46 string Laptop::getManufacturer() const
47 {
48     return manufacturer;
```

```
49 }
50 string Laptop::getProcessor() const
51 {
52     return processor;
53 }
54 double Laptop::getScreenSize() const
55 {
56     return screenSize;
57 }
58 string Laptop::getColor() const
59 {
60     return color;
61 }
62 bool Laptop::getIsNew() const
63 {
64     return isNew;
65 }
66 void Laptop::setManufacturer(string x)
67 {
68     manufacturer = x;
69 }
70 void Laptop::setProcessor(string x)
71 {
72     processor = x;
73 }
74 void Laptop::setColor(string x)
75 {
76     color = x;
77 }
78 void Laptop::setScreenSize(double x)
79 {
80     screenSize = x;
81 }
82 void Laptop::setIsNew(bool x)
83 {
84     isNew = x;
85 }
86 void Laptop::input()
87 {
88     string newStr;
89     cout << "Manufacturer? ";
90     getline(cin, manufacturer);
91     cout << "Processor? ";
92     getline(cin, processor);
93     cout << "ScreenSize? ";
94     cin >> screenSize;
95     cin.ignore();
96     cout << "Color? ";
97     getline(cin, color);
```

```
198     cout << "Is this laptop new?(Y/N) ";
199     getline(cin, newStr);
200     if (newStr == "Y" || newStr == "y")
201         isNew = true;
202     else if (newStr == "N" || newStr == "n")
203         isNew = false;
204     else
205         cout << "Invalid answer!!!" << endl;
206     cout << endl;
207 }
208 void Laptop::output() const
209 {
210     cout << setw(10) << manufacturer << setw(15) << processor << "\t" <<
        screenSize << "\t" << color << "\t";
211     if(isNew)
212         cout << "New" << endl;
213     else
214         cout << "Used" << endl;
215 }
216 bool operator==(const Laptop& a, const Laptop& b)
217 {
218     return (a.manufacturer == b.manufacturer) &&
219           (a.processor == b.processor) &&
220           (a.screenSize == b.screenSize) &&
221           (a.color == b.color) &&
222           (a.isNew == b.isNew);
223 }
224 class ElectronicStore
225 {
226 private:
227     string storeName;
228     string city;
229     int numLaptops;
230     int maxLaptops;
231     Laptop* list;
232 public:
233     ElectronicStore(int);
234     ~ElectronicStore();
235     string getStoreName() const;
236     string getCity() const;
237     int getNumLaptops() const;
238     int getMaxLaptops() const;
239     void setStoreName(string);
240     void setCity(string);
241     void addLaptop(Laptop);
242     void sellLaptop(Laptop);
243     void displayAll();
244     void displayBrand(string);
245 };
```

```
146 ElectronicStore::ElectronicStore(int s)
147 {
148     storeName = "Best Buy";
149     city = "Vancouver";
150     numLaptops = 0;
151     maxLaptops = s;
152     list = new Laptop[maxLaptops];
153 }
154 ElectronicStore::~ElectronicStore()
155 {
156     delete[] list;
157 }
158 string ElectronicStore::getStoreName() const
159 {
160     return storeName;
161 }
162 string ElectronicStore::getCity() const
163 {
164     return city;
165 }
166 int ElectronicStore::getNumLaptops() const
167 {
168     return numLaptops;
169 }
170 int ElectronicStore::getMaxLaptops() const
171 {
172     return maxLaptops;
173 }
174 void ElectronicStore::setStoreName(string x)
175 {
176     storeName = x;
177 }
178 void ElectronicStore::setCity(string x)
179 {
180     city = x;
181 }
182 void ElectronicStore::addLaptop(Laptop x)
183 {
184     if (numLaptops < maxLaptops)
185     {
186         list[numLaptops] = x;
187         numLaptops++;
188     }
189     else
190         cout << "No more room for a new laptop\n";
191 }
192 void ElectronicStore::sellLaptop(Laptop x)
193 {
194     int i;
```

```
195     for (i = 0; i < numLaptops; i++)
196     {
197         if (list[i] == x)
198         {
199             list[i] = list[numLaptops - 1];
200             numLaptops--;
201             return;
202         }
203     }
204     cout << "Sorry! This laptop is not in stock!!! \n";
205 }
206
207 void ElectronicStore::displayAll()
208 {
209     cout << "Welcome to " << storeName << " " << city << endl;
210     cout << "This store has " << numLaptops << " laptops list below: \n"
211         << endl;
212     for (int i = 0; i < numLaptops; i++)
213         list[i].output();
214     cout << endl;
215 }
216 void ElectronicStore::displayBrand(string s)
217 {
218     cout << "These are " << s << " laptops we have in stock:\n";
219     for (int i = 0; i < numLaptops; i++)
220         if (list[i].getManufacturer() == s)
221             list[i].output();
222     cout << endl;
223 }
224 int main()
225 {
226     ElectronicStore x(1000), y(100);
227     x.setStoreName("Visions");
228     x.setCity("Burnaby");
229     y.setStoreName("Joe's computer");
230     y.setCity("Maple Ridge");
231
232     Laptop a("Dell", "Intel i7", 17.3, "Silver", true), b("hp", "AMD",
233         17.3, "White", false), c("Acer");
234     x.addLaptop(a);
235     x.addLaptop(a);
236     x.addLaptop(a);
237     x.addLaptop(b);
238     x.addLaptop(b);
239     x.addLaptop(c);
240     x.displayAll();
241 }
```

---

```
242     x.sellLaptop(b);
243     x.displayAll();
244
245     x.displayBrand("Dell");
246     return 0;
247 }
248
```