

8.19 Problem set

Problem 8.19.1. Implement *employee* class as explained in figure 8.29. The driver program *test.cpp* is given. All the tests must pass. The program must use a static data member called *show* in the *employee* class. When this variable is *true*, the program must print the call of constructors, destructors, copy constructors and equal operators.

email the following files, along with copy of the output of the program.

```
employee.h
employee.cpp
salariedemployee.h
salariedemployee.cpp
commissionemployee.h
commissionemployee.cpp
basepluscommissionemployee.h
basepluscommissionemployee.cpp
output of the program as a doc file
```

```
1  /*-----
2  Copyright (c) 2018 Author: Jagadeesh Vasudevamurthy
3  Filename: test.cpp
4
5  Memory leaked: 0 bytes (0%); potentially leaked: 0 bytes (0%)
6  -----*/
7
8  /*-----
9          NOTHING CAN BE CHANGED BELOW
10 -----*/
11
12 /*-----
13 All includes here
14 -----*/
15 #include "employee.h"
16 #include "salariedemployee.h"
17 #include "commissionemployee.h"
18 #include "basepluscommissionemployee.h"
19
20 /*-----
21 testEmployee
22 -----*/
23 void testEmployee() {
24     employee e1("jag", "vasudevamurthy", "678-90");
25     cout << e1 << endl;
26     employee e2("bob", "ericission", "178-90");
27     cout << e2 << endl;
28     employee e3(e1);
29     cout << e3 << endl;
30     e2 = e3;
31     cout << e3 << endl;
32     cout << e2 << endl;
33 }
34
35 /*-----
36 testSalariedemployee
37 -----*/
38 void testSalariedemployee() {
39     salariedemployee e1("jag", "vasudevamurthy", "678-90", "Xilinx", 56000);
40     cout << e1 << endl;
41     salariedemployee e2("bob", "ericission", "178-90", "ATT", 3456000);
42     cout << e2 << endl;
43     salariedemployee e3(e1);
44     cout << e3 << endl;
45     e2 = e3;
46     cout << e3 << endl;
47     cout << e2 << endl;
48 }
49
50 /*-----
51 test Commission employee
52 -----*/
```

```
53 void testCommissionemployee() {
54     commissionemployee e1("jag", "vasudevamurthy", "678-90", "BELL", 560);
55     cout << e1 << endl;
56     commissionemployee e2("bob", "ericission", "178-90", "UCSC", 345);
57     cout << e2 << endl;
58     commissionemployee e3(e1);
59     cout << e3 << endl;
60     e2 = e3;
61     cout << e3 << endl;
62     cout << e2 << endl;
63 }
64
65 /*-----
66 test Basepluscommissionemployee
67 -----*/
68 void testBasepluscommissionemployee() {
69     basepluscommissionemployee e1("jag", "vasudevamurthy", "678-90", "BELL",
560, "abc", 89);
70     cout << e1 << endl;
71     basepluscommissionemployee e2("bob", "ericission", "178-90", "UCSC", 345,
"xyz", 90);
72     cout << e2 << endl;
73     basepluscommissionemployee e3(e1);
74     cout << e3 << endl;
75     e2 = e3;
76     cout << e3 << endl;
77     cout << e2 << endl;
78 }
79
80 /*-----
81 test
82 -----*/
83 void polymorphism() {
84     vector<employee*> v;
85     {
86         employee* e1 = new employee("jag", "vasudevamurthy", "678-90");
87         v.push_back(e1);
88         employee* e2 = new employee("bob", "ericission", "178-90");
89         v.push_back(e2);
90     }
91     {
92         salariedemployee* e1 = new salariedemployee("alex", "wong", "678-89",
"Men", 24);
93         v.push_back(e1);
94         salariedemployee* e2 = new salariedemployee("bob", "john", "478-90",
"XYX", 345);
95         v.push_back(e2);
96     }
97
98     {
99         commissionemployee* e1 = new commissionemployee("hale", "tom", "589-78",
, "xil", 560000);
```

```
100     v.push_back(e1);
101     commissionemployee* e2 = new commissionemployee("jon", "snith",
102     "478-90", "UCB", 9);
103     v.push_back(e2);
104 }
105 {
106     basepluscommissionemployee* e1 = new basepluscommissionemployee("amy",
107     "vmay", "878-90", "synp", 160, "nvida", 809);
108     v.push_back(e1);
109     basepluscommissionemployee* e2 = new basepluscommissionemployee("fake",
110     "guy", "878-60", "mu", 345, "spy", 190);
111     v.push_back(e2);
112 }
113 {
114     for (auto i = 0; i < v.size(); ++i) {
115         cout << i << ": " << *(v[i]) << endl;
116     }
117 }
118 {
119     for (auto i : v) {
120         delete i;
121     }
122 }
123
124
125 /*-----
126 -----*/
127
128 int main() {
129     #ifdef _WIN32
130     _CrtSetDbgFlag(_CRTDBG_ALLOC_MEM_DF | _CRTDBG_LEAK_CHECK_DF);
131     #endif
132     employee::setShow(true);
133     testEmployee();
134     testSalariedemployee();
135     testCommissionemployee();
136     testBasepluscommissionemployee();
137     polymorphism();
138     return 0;
139 }
```

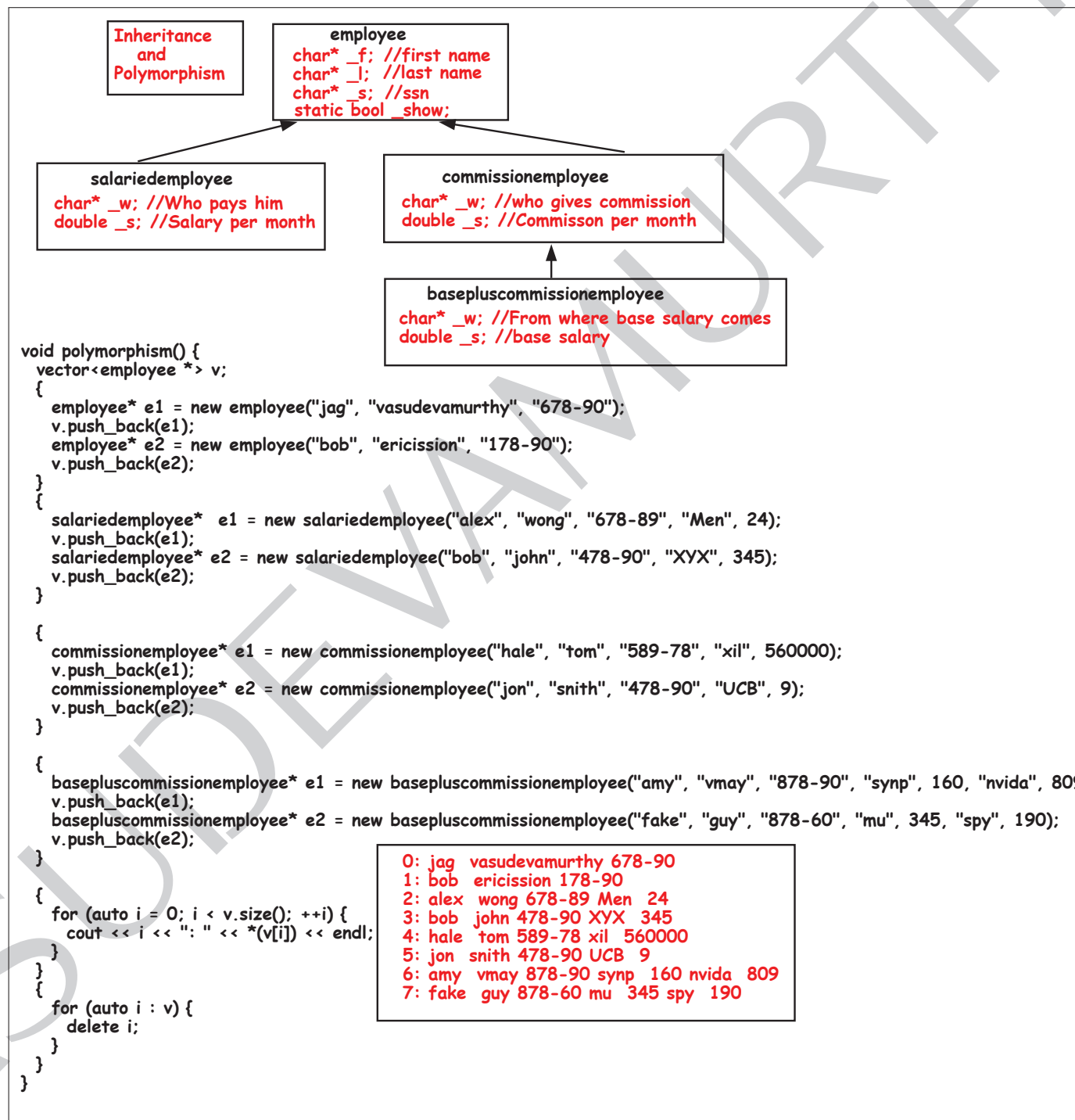


Figure 8.29: Implementing inheritance and polymorphism