```cpp
// Justin Dang Student ID: 1148267
/*
Creates a base class containing common classes and data members. includes pure virtual functions.

Derived classes represent each package type and include calculations specified for each package type

An additional package type is included with specified prices
*/
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <string>
#include <iomanip>
using namespace std;

class Customer {
public:
    Customer(const char *, const char *, const char *, const char *, int);
    ~Customer();
    virtual float calcDiscountPerc(float) const;        // total amount
    virtual float calcDiscountAmnt(float, float) const; // (total amount, discount percent)
    virtual float calcShipCost(int) const = 0;
    virtual float calcOD(bool, float) const = 0;
    virtual float calcInsurance(bool, float) const = 0;
    virtual void print() const;

private:
    char* name, * address, * city, * state;
    int zip;
};
Customer::Customer(const char* n, const char* a, const char* c, const char* s,
    int z) {
    name = new char[strlen(n) + 1];
    address = new char[strlen(a) + 1];
    city = new char[strlen(c) + 1];
    state = new char[strlen(s) + 1];

    strcpy(name, n);
    strcpy(address, a);
    strcpy(city, c);
    strcpy(state, s);

    zip = z;
}
float Customer::calcDiscountPerc(float t) const{
    if (t <= 300.00)
        return (float)0.019;
    else if (t > 300.00 && t <= 600.00)
        return (float)0.026;
    else if (t > 600.00 && t <= 1000.00)
```

```
50          return (float)0.029;
51      else if (t > 1000.00)
52          return (float)0.034;
53      else
54          return (float)0;
55  }
56  // t = total cost, p = percent discount
57  float Customer::calcDiscountAmnt(float t, float p) const { return t * p; }
58  void Customer::print() const{
59      cout << endl << "Customer:    " << setw(1) << left << name << setw(45) <<
          right << "Expected Arrival Date:  8-25-2018" << endl;
60      cout << setw(28) << address << endl;
61      cout << setw(18) << city << ", " << state << ' ' << zip;
62      cout << "\n\n";
63  }
64  Customer::~Customer() {
65      delete[] name, address, city, state;
66  }
67
68  class Games : public Customer {
69  public:
70      Games(const char* , const char* , const char* , const char* ,
71          int , int, bool, bool); // games(units,false or true for overnight
              delivery
72                              // ,false or true for insurance)
73      virtual float calcShipCost(int) const;
74      virtual float calcOD(bool, float) const; // (total amount)
75      virtual float calcInsurance(bool, float) const; // total amount
76      virtual void print() const;
77  private:
78      int units;
79      bool od, insurance;
80      float total, sCost, d, iCost, dPerc, dAmnt;
81
82  };
83  Games::Games(const char* n, const char* a, const char* c, const char* s,
84      int z, int u, bool o, bool i) : Customer(n, a, c, s, z)
85  {
86      units = u;
87      od = o;
88      insurance = i;
89      sCost = calcShipCost(units);
90      d = calcOD(od, sCost);
91      iCost = calcInsurance(insurance, sCost);
92      total = (sCost + d + iCost);
93      dPerc = Customer::calcDiscountPerc(total);
94      dAmnt = Customer::calcDiscountAmnt(total, dPerc);
95  }
96  float Games::calcShipCost(int u) const { return (float)u * (float)19.99; }
97  float Games::calcOD(bool o, float t) const { return o ? (float)4.99 : (float)
      0.0; }
98  float Games::calcInsurance(bool i, float t) const { return i ? ((float)t *
```

```
      (float)0.06) : (float)0.0; }
 99
100 void Games::print() const{
101     Customer::print();
102     cout << showpoint << fixed << setprecision(2);
103     cout << units << " Video Game(s) ordered: shipping cost is $" << sCost;
104     if (od) {
105         cout << ", overnight delivery is $4.99";
106     }
107     else
108         cout << ", normal delivery preferred";
109     if (insurance)
110         cout << ", insurance cost is $" << iCost;
111     else
112         cout << ", no insurance specified";
113     cout << endl << "Total Cost is $" << total;
114     cout << endl << "Discount Percentage is " << dPerc * 100 <<
115         "% for a discount of $" << dAmnt;
116     cout << endl << "total Cost after discount is $" << total - dAmnt;
117 }
118
119 class Phones : public Customer {
120 public:
121     Phones(const char*, const char*, const char*, const char*,
122         int, int, int, bool, bool); // games(units,false or true for overnight
          delivery
123                                     // ,false or true for insurance)
124     virtual float calcShipCost(int) const;
125     virtual float calcOD(bool, float) const; // (total amount)
126     virtual float calcInsurance(bool, float) const; // total amount
127     float calcPCCost(int);
128     virtual void print() const;
129 private:
130     int units, pCaseUnits;
131     bool od, insurance;
132     float total, sCost, d, iCost, dPerc, dAmnt, pCCost;
133 };
134 Phones::Phones(const char* n, const char* a, const char* c, const char* s,
135     int z, int u, int pCU, bool o, bool i) : Customer(n, a, c, s, z)
136 {
137     units = u;
138     pCaseUnits = pCU;
139     od = o;
140     insurance = i;
141     sCost = calcShipCost(units);
142     d = calcOD(od, sCost);
143     pCCost = calcPCCost(pCU);
144     iCost = calcInsurance(insurance, sCost);
145     total = (sCost + d + iCost + pCCost);
146     dPerc = Customer::calcDiscountPerc(total);
147     dAmnt = Customer::calcDiscountAmnt(total, dPerc);
148 }
```

```
149 float Phones::calcShipCost(int u) const { return (float)u * (float)799.99; }
150 float Phones::calcOD(bool o, float t) const { return o ? t + (float)25.00 :
      (float)0.0; }
151 float Phones::calcInsurance(bool i, float t) const { return i ? (t * (float)
      0.11) : (float)0.0; }
152 float Phones::calcPCCost(int u) { return (float)u * (float)24.99; }
153 void Phones::print() const {
154     Customer::print();
155     cout << showpoint << fixed << setprecision(2);
156     cout << units << " Genius Phone(s) ordered: shipping cost is $" << sCost;
157
158     od ? cout << ", overnight delivery is $4.99" : cout << ", normal delivery
        preferred";
159     insurance ? cout << ", insurance cost is $" << iCost : cout << ", no
        insurance specified";
160     pCaseUnits > 0 ? cout << ", " << pCaseUnits << " phone case ordered for $" <<
          pCCost :
161         cout << ", No phone cases ordered";
162
163     cout << endl << "Total Cost is $" << total;
164     cout << endl << "Discount Percentage is " << dPerc * 100 << "% for a discount
        of $" << dAmnt;
165     cout << endl << "total Cost after discount is $" << total - dAmnt;
166 }
167 // ____________________________________________________________
168 class Hamburgers : public Customer {
169 public:
170     Hamburgers(const char*, const char*, const char*, const char*,
171         int, int, int, int, bool, bool); // games(units,false or true for
              overnight delivery
172                                     // ,false or true for insurance)
173     virtual float calcShipCost(int) const;
174     virtual float calcOD(bool, float) const; // (total amount)
175     virtual float calcInsurance(bool, float) const; // total amount
176     float calcCCost(int);
177     float calcBCost(int);
178     virtual void print() const;
179 private:
180     int units, cAmnt, bAmnt;
181     bool od, insurance;
182     float total, sCost, d, iCost, dPerc, dAmnt, cCost, bCost;
183 };
184 Hamburgers::Hamburgers(const char* n, const char* a, const char* c, const char*
      s,
185     int z, int u, int cU, int bU, bool o, bool i) : Customer(n, a, c, s, z)
186 {
187     units = u;
188     cAmnt = cU;
189     bAmnt = bU;
190     od = o;
191     insurance = i;
192     sCost = calcShipCost(units);
```

```
193     bCost = calcBCost(bAmnt);
194     d = calcOD(od, sCost);
195     cCost = calcCCost(cAmnt);
196     iCost = calcInsurance(insurance, sCost);
197     total = (sCost + d + iCost + cCost + bCost);
198     dPerc = Customer::calcDiscountPerc(total);
199     dAmnt = Customer::calcDiscountAmnt(total, dPerc);
200 }
201 float Hamburgers::calcShipCost(int u) const { return (float)u * (float)7.99; }
202 float Hamburgers::calcOD(bool o, float t) const { return o ? t + (float)9.00 :
      (float)0.0; }
203 float Hamburgers::calcInsurance(bool i, float t) const { return i ? (t * (float)
      0.03) : (float)0.0; }
204 float Hamburgers::calcBCost(int u) { return (float)u * (float)1.89; }
205 float Hamburgers::calcCCost(int u) { return (float)u * (float)1.79; }
206 void Hamburgers::print() const {
207     Customer::print();
208     cout << showpoint << fixed << setprecision(2);
209     cout << units << " Hamburger(s) ordered: shipping cost is $" << sCost;
210
211     od ? cout << ", overnight delivery is $4.99" : cout << ", normal delivery
        preferred";
212     insurance ? cout << ", insurance cost is $" << iCost : cout << ", no
        insurance specified";
213     cAmnt > 0 ? cout << ", " << cAmnt << " condiments ordered for $" << cCost :
214         cout << ", No condiments ordered";
215     bAmnt > 0 ? cout << ", " << bAmnt << " buns ordered for $" << bCost :
216         cout << ", No buns ordered";
217
218     cout << endl << "Total Cost is $" << total;
219     cout << endl << "Discount Percentage is " << dPerc * 100 << "% for a discount
        of $" << dAmnt;
220     cout << endl << "total Cost after discount is $" << total - dAmnt;
221 }
222 // ____________________________________________
223 /*
224 soft cover textbooks - 199.99
225 hard cover textbooks - 100.00
226 answer key - 49.99
227 overnight - 14.99
228 insurance - 5%
229 */
230 class TextBooks : public Customer {
231 public:
232     TextBooks(const char*, const char*, const char*, const char*,
233         int, int, int, int, bool, bool); // games(units,false or true for
            overnight delivery
234                             // ,false or true for insurance)
235     virtual float calcShipCost(int) const;
236     virtual float calcOD(bool, float) const; // (total amount)
237     virtual float calcInsurance(bool, float) const; // total amount
238     virtual void print() const;
```

```
239     float calcAC(int) const;
240     float calcHC(int) const;
241 private:
242     int units, aKUnit, hCUnit;
243     bool od, insurance;
244     float total, sCost, d, iCost, dPerc, dAmnt, aKCost, hCCost;
245
246 };
247 TextBooks::TextBooks(const char* n, const char* a, const char* c, const char* s,
248     int z, int u, int aKU, int hCU, bool o, bool i) : Customer(n, a, c, s, z)
249 {
250     units = u;
251     od = o;
252     insurance = i;
253     aKUnit = aKU;
254     hCUnit = hCU;
255     sCost = calcShipCost(units);
256     d = calcOD(od, sCost);
257     iCost = calcInsurance(insurance, sCost);
258     aKCost = calcAC(aKUnit);
259     hCCost = calcHC(hCUnit);
260     total = (sCost + d + iCost + aKCost + hCCost);
261     dPerc = Customer::calcDiscountPerc(total);
262     dAmnt = Customer::calcDiscountAmnt(total, dPerc);
263 }
264 float TextBooks::calcShipCost(int u) const { return (float)u * (float)199.99; }
265 float TextBooks::calcOD(bool o, float t) const { return o ? (float)14.99 :
      (float)0.0; }
266 float TextBooks::calcInsurance(bool i, float t) const { return i ? ((float)t *
      (float)0.05) : (float)0.0; }
267 float TextBooks::calcAC(int u) const { return (float)u * (float)49.99; }
268 float TextBooks::calcHC(int u) const { return (float)u * (float)100.00; }
269 void TextBooks::print() const {
270     Customer::print();
271     cout << showpoint << fixed << setprecision(2);
272     cout << units << " Textbook(s) ordered: shipping cost is $" << sCost;
273
274     od ? cout << ", overnight delivery is $4.99" : cout << ", normal delivery
        preferred";
275     insurance ? cout << ", insurance cost is $" << iCost : cout << ", no
        insurance specified";
276     hCUnit > 0 ? cout << ", " << hCUnit << " hard cover textbooks ordered for $"
        << hCCost :
277         cout << ", no hard cover textbooks ordered";
278     aKUnit > 0 ? cout << ", " << aKUnit << " answer keys ordered for $" <<
        aKCost :
279         cout << ", no answer keys ordered";
280
281     cout << endl << "Total Cost is $" << total;
282     cout << endl << "Discount Percentage is " << dPerc * 100 <<
283         "% for a discount of $" << dAmnt;
284     cout << endl << "total Cost after discount is $" << total - dAmnt;
```

```
285 }
286 int main()
287 {
288     Customer* ptr;
289     ptr = new Games("Joe Blow", "1234 Main Street", "Irvine", "CA", 92618, 4,
          true, false);
290     ptr->print();
291
292     cout << endl;
293     ptr = new Phones("Joe Blow", "1234 Main Street", "Irvine", "CA", 92618, 8, 4,
           false, true);
294     ptr->print();
295
296     cout << endl;
297     ptr = new Hamburgers("Joe Blow", "1234 Main Street", "Irvine", "CA", 92618,
          16, 16, 32, false, false);
298     ptr->print();
299
300     cout << endl;
301     ptr = new TextBooks("Joe Blow", "1234 Main Street", "Irvine", "CA", 92618, 1,
           1, 0, false, false);
302     ptr->print();
303 }
```