```
/*  Program: A5P2 - Pairs
    Author: Tom Stutler
    Last Date Modified: 5/7/2015

    The intent of this program is to demonstrate knowledge
    of inherited classes in c++.
*/

#include <iostream>
#include <cmath>

using namespace std;

const int SZ=6;

class pairs
{
protected:
    float a, b;

public:
    pairs () : a(0), b(0) {}
    pairs (const pairs& p) : a(p.a), b(p.b) {}

    bool operator ==(pairs& pairsParam)
        {
            if (a==pairsParam.a && b==pairsParam.b) {
                return true;
            } else {
                return false;
            }
        }
    pairs operator +(pairs& pairsParam)
        {
            pairs newPair;
            newPair.a = a+pairsParam.a;
            newPair.b = b+pairsParam.b;
            return newPair;
        }
    pairs operator -(pairs& pairsParam)
        {
            pairs newPair;
            newPair.a = a-pairsParam.a;
            newPair.b = b-pairsParam.b;
            return newPair;
        }
};

class complex : public pairs
{
private:
    char oper;

public:
    complex () : oper('+') {pairs();}
    complex (const pairs& p) : pairs(p) {}

    complex operator *(complex& cParam)
    {
        complex newComplex;
        newComplex.a = (a*cParam.a) - (b*cParam.b);
        newComplex.b = (a*cParam.b) + (b*cParam.a);
        return newComplex;
    }
    complex operator /(complex& cParam)
    {
```

```
 67          complex newComplex;
 68          newComplex.a = (a*cParam.a + b*cParam.b)/(pow(cParam.a, 2) + pow(cParam.b,
2));
 69          newComplex.b = (a*cParam.a - b*cParam.b)/(pow(cParam.a, 2) + pow(cParam.b,
2));
 70          return newComplex;
 71      }
 72
 73      friend ostream& operator <<(ostream& output, complex& cParam)
 74      {
 75          if (cParam.b>=0) {
 76              output << cParam.a << cParam.oper << cParam.b << "i";
 77              return output;
 78          } else {
 79              output << cParam.a << cParam.b << "i";
 80              return output;
 81          }
 82      }
 83      friend istream& operator >>(istream& input, complex& cParam)
 84      {
 85          char i;
 86          input >> cParam.a >> cParam.oper >> cParam.b >> i;
 87          if (cParam.oper == '-') {
 88              cParam.b = 0-cParam.b;
 89          }
 90          return input;
 91      }
 92  };
 93
 94  class vect : public pairs
 95  {
 96  public:
 97      vect () {pairs();}
 98      vect (const pairs& p) : pairs(p) {}
 99
100      float operator *(vect& vParam)
101      {
102          return (a*vParam.a + b*vParam.b);
103      }
104      vect operator *(float numParam)
105      {
106          vect newVect;
107          newVect.a = a*numParam;
108          newVect.b = b*numParam;
109          return newVect;
110      }
111
112      friend ostream& operator <<(ostream& output, vect& vParam)
113      {
114          output << "<" << vParam.a << "," << vParam.b << ">";
115          return output;
116      }
117      friend istream& operator >>(istream& input, vect& vParam)
118      {
119          char bracket, comma;
120          input >> bracket >> vParam.a >> comma >> vParam.b >> bracket;
121          return input;
122      }
123  };
124
125  int selectset();
126  void ComplexLand (complex cArry[]);
127  void VectorLand (vect vArry[]);
128
129  int main()
130  {
```

```
131     int sel;
132     complex cx[SZ];
133     vect vx[SZ];
134 
135     do {
136         cout << "Select an option - (1) Enter Complex Number Land\n";
137         cout << "                   (2) Enter Vector Land\n";
138         cout << "                   (3) Exit\n";
139         cin >> sel;
140         if(sel == 1)
141             ComplexLand(cx);
142         else if(sel == 2)
143             VectorLand(vx);
144         else if (sel == 3)
145             cout << "Bye...\n";
146         else
147             cout << "Invalid input - try again\n";
148     } while (sel != 3);
149 
150     char ch; cin >> ch;
151 return 0;
152 }
153 
154 int selectset()
155 {
156     int iset;
157     char set;
158 
159     do
160     {
161         cout << "(enter A-F): ";
162         cin >> set;
163         set = toupper(set);
164         iset = set-'A';
165         if (iset<0 || iset>5) {
166             cout << "Invalid - reenter\n";
167         }
168     } while (iset<0 || iset>5);
169 
170     return iset;
171 }
172 
173 void ComplexLand (complex cArry[])
174 {
175     int selMenu, set1, set2, set3;
176     char selOper;
177     complex temp;
178 
179     do
180     {
181         cout << "Select an option:\n"
182              << "\t(1) Enter a complex number\n"
183              << "\t(2) Display all complex numbers\n"
184              << "\t(3) Perform arithmetic or equality check\n"
185              << "\t(4) Exit\n";
186         cin >> selMenu;
187 
188         switch (selMenu)
189         {
190         case 1:
191             cout << "Enter a complex number in a+bi or a-bi form: ";
192             cin >> temp;
193             cout << "Where do you want to store this ";
194             set1 = selectset();
195             cArry[set1] = temp;
196             break;
```

```
197         case 2:
198             cout << "A: " << cArry[0]
199                  << "\nB: " << cArry[1]
200                  << "\nC: " << cArry[2]
201                  << "\nD: " << cArry[3]
202                  << "\nE: " << cArry[4]
203                  << "\nF: " << cArry[5] << endl;
204                  break;
205         case 3:
206             cout << "Enter an operation:   +   -   *   /   =  ";
207             cin >> selOper;
208 
209             if (selOper=='=') {
210                 cout << "Enter first number ";
211                 set1 = selectset();
212                 cout << "Enter second number ";
213                 set2 = selectset();
214 
215                 if (cArry[set1] == cArry[set2]) {
216                     cout << cArry[set1] << " does equal " << cArry[set2] << endl;
217                 } else {
218                     cout << cArry[set1] << " does not equal " << cArry[set2] <<
endl;
219                 }
220             } else if (selOper=='/') {
221                 cout << "Enter first number ";
222                 set1 = selectset();
223                 cout << "Enter second number ";
224                 set2 = selectset();
225                 cout << "Enter result location ";
226                 set3 = selectset();
227 
228                 cArry[set3] = cArry[set1]/cArry[set2];
229             } else if (selOper=='*') {
230                 cout << "Enter first number ";
231                 set1 = selectset();
232                 cout << "Enter second number ";
233                 set2 = selectset();
234                 cout << "Enter result location ";
235                 set3 = selectset();
236 
237                 cArry[set3] = cArry[set1]*cArry[set2];
238             } else if (selOper=='-') {
239                 cout << "Enter first number ";
240                 set1 = selectset();
241                 cout << "Enter second number ";
242                 set2 = selectset();
243                 cout << "Enter result location ";
244                 set3 = selectset();
245 
246                 cArry[set3] = cArry[set1]-cArry[set2];
247             } else if (selOper=='+') {
248                 cout << "Enter first number ";
249                 set1 = selectset();
250                 cout << "Enter second number ";
251                 set2 = selectset();
252                 cout << "Enter result location ";
253                 set3 = selectset();
254 
255                 cArry[set3] = cArry[set1]+cArry[set2];
256             } else {
257                 cout << "Invalid input.\n";
258             }
259             break;
260         default:
261             cout << "Invalid input. Please try again.\n";
```

```
262             break;
263
264         }
265     }while (selMenu != 4);
266 }
267
268 void VectorLand (vect vArry[])
269 {
270     int selMenu, set1, set2, set3;
271     float scalar;
272     char selOper;
273     vect temp;
274
275     do
276     {
277         cout << "Select an option:\n"
278              << "\t(1) Enter a complex number\n"
279              << "\t(2) Display all complex numbers\n"
280              << "\t(3) Perform arithmetic or equality check\n"
281              << "\t(4) Exit\n";
282         cin >> selMenu;
283
284         switch (selMenu)
285         {
286         case 1:
287             cout << "Enter a complex number in a+bi or a-bi form: ";
288             cin >> temp;
289             cout << "Where do you want to store this ";
290             set1 = selectset();
291             vArry[set1] = temp;
292             break;
293         case 2:
294             cout << "A: " << vArry[0]
295                  << "\nB: " << vArry[1]
296                  << "\nC: " << vArry[2]
297                  << "\nD: " << vArry[3]
298                  << "\nE: " << vArry[4]
299                  << "\nF: " << vArry[5] << endl;
300                  break;
301         case 3:
302             cout << "Enter an operation:   +   -   .(dot prod)   *(scalar prod)   =
";
303             cin >> selOper;
304
305             if (selOper=='=') {
306                 cout << "Enter first number ";
307                 set1 = selectset();
308                 cout << "Enter second number ";
309                 set2 = selectset();
310
311                 if (vArry[set1] == vArry[set2]) {
312                     cout << vArry[set1] << " does equal " << vArry[set2] << endl;
313                 } else {
314                     cout << vArry[set1] << " does not equal " << vArry[set2] <<
endl;
315                 }
316             } else if (selOper=='.') {
317                 cout << "Enter first number ";
318                 set1 = selectset();
319                 cout << "Enter second number ";
320                 set2 = selectset();
321                 cout << "The dot product = " << vArry[set1]*vArry[set2] << endl;
322             } else if (selOper=='*') {
323                 cout << "Enter first number ";
324                 set1 = selectset();
325                 cout << "Enter scalar: ";
```

```
                cin >> scalar;
                cout << "Enter result location ";
                set2 = selectset();
                vArry[set2] = vArry[set1]*scalar;
            } else if (selOper=='-') {
                cout << "Enter first number ";
                set1 = selectset();
                cout << "Enter second number ";
                set2 = selectset();
                cout << "Enter result location ";
                set3 = selectset();

                vArry[set3] = vArry[set1]-vArry[set2];
            } else if (selOper=='+') {
                cout << "Enter first number ";
                set1 = selectset();
                cout << "Enter second number ";
                set2 = selectset();
                cout << "Enter result location ";
                set3 = selectset();

                vArry[set3] = vArry[set1]+vArry[set2];
            } else {
                cout << "Invalid input.\n";
            }
            break;
        default:
            cout << "Invalid input. Please try again.\n";
            break;

        }
    }while (selMenu != 4);
}
```