

```

1  /* Program: A4P3 - dynarray
2     Author: Tom Stutler
3     Last Date Modified: 4/8/15
4
5     The intent of this program is to create a class, dynarray, to be
6     used in conjunction with a provided main() and myfunc() function.
7  */
8
9  #include <iostream>
10
11  using namespace std;
12
13  class dynarray
14  {
15  public:
16      dynarray() : maxSize(0), usedSize(0)
17          { intPtr = new int[maxSize]; }
18      dynarray(int sizeParam) : maxSize(sizeParam), usedSize(0)
19          { intPtr = new int[maxSize]; }
20
21      void show (int indexParam);
22      void set  (int indexParam, int valueParam);
23      void expand (int sizeParam);
24
25      ~dynarray();
26
27  private:
28      int  maxSize;
29      int  usedSize;
30      int  *intPtr;
31
32  };
33
34  void myfunc();
35
36  int main()
37  {
38      int size, more, i;
39      dynarray y;
40
41      cout << "Enter dynamic array size: ";
42      cin >> size;
43      dynarray x(size);
44
45      for(i=0; i<size; i++)
46          x.set(i, 3*i);
47
48      for(i=0; i<size; i++)
49          x.show(i);
50
51      cout << "How much more dynamic array space do you want? ";
52      cin >> more;
53      x.expand(more);
54
55      for(i=0; i<(size+more); i++)
56          x.set(i, 5*i);
57
58      for(i=0; i<(size+more); i++)
59          x.show(i);
60
61      x.show(size+more+5); //invalid index in show
62      x.set(-2, 9); //invalid index in set
63      y.set(3, 6); //empty dynarray set
64      y.show(3); //empty dynarray show
65      myfunc();
66

```

```

67     return 0;
68 }
69
70 void myfunc()
71 {
72     int i;
73
74     cout << "hi from myfunc...\n";
75     dynarray y(5);
76
77     for(i=0; i<5; i++)
78         y.set(i, i*i);
79
80     for(i=0; i<5; i++)
81         y.show(i);
82
83     cout << "bye from myfunc...\n";
84 }
85
86 void dynarray::show (int indexParam)
87 {
88     if (maxSize==0) {
89         cout << "Cannot show - dynarray empty\n";
90     } else if (indexParam>usedSize || indexParam<0) {
91         cout << "Invalid index in show\n";
92     } else {
93         cout << *(intPtr+indexParam) << endl;
94     }
95 }
96
97 void dynarray::set (int indexParam, int valueParam)
98 {
99     if (maxSize==0) {
100         cout << "Cannot set - dynarray empty\n";
101     } else if (indexParam>maxSize || indexParam<0) {
102         cout << "Invalid index in set\n";
103     } else {
104         *(intPtr+indexParam) = valueParam;
105         usedSize++;
106     }
107 }
108
109 void dynarray::expand (int sizeParam)
110 {
111     int *temp = new int[maxSize+sizeParam];
112
113     for (int i=0; i<usedSize; i++) {
114         *(temp+i) = *(intPtr+i);
115     }
116
117     delete [] intPtr;
118     intPtr = temp;
119 }
120
121 dynarray::~dynarray()
122 {
123     cout << "hi from the dynarray destructor...\n";
124     delete [] intPtr;
125 }

```