**Chapter 9 Inserts**

**Insert 535-A**

1 // This program demonstrates a function that returns a pointer.  
 2 #include <iostream>  
 3 #include <random>  
 4 using namespace std;  
 5   
 6 // Function prototype  
 7 int \*getRandomNumbers(int);  
 8   
 9 int main()  
10 {  
11 int\* numbers; // To point to the numbers  
12   
13 // Get an array of five random numbers.  
14 numbers = getRandomNumbers(5);  
15   
16 // Display the numbers.  
17 for (int count = 0; count < 5; count++)  
18 cout << numbers[count] << endl;  
19   
20 // Free the memory.  
21 delete[] numbers;  
22 numbers = nullptr;  
23 return 0;  
24 }  
25   
26 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
27 // The getRandomNumbers function returns a pointer to an \*  
28 // array of random integers. The num parameter is the \*  
29 // number of numbers requested. \*  
30 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
31   
32 int \*getRandomNumbers(int num)  
33 {  
34 const int MIN = 0; // Minimum random number  
35 const int MAX = 100; // Maximum random number  
36 int\* arr = nullptr; // Array to hold the numbers  
37   
38 // Random number engine and distribution object  
39 random\_device engine;  
40 uniform\_int\_distribution<int> randInt(MIN, MAX);  
41   
42 // Return null if num is zero or negative.  
43 if (num <= 0)  
44 return nullptr;  
45   
46 // Dynamically allocate the array.  
47 arr = new int[num];  
48   
49 // Populate the array with random numbers.  
50 for (int count = 0; count < num; count++)  
51 arr[count] = randInt(engine);  
52   
53 // Return a pointer to the array.  
54 return arr;  
55 }

**Program Output**

51

11

85

16

33