



Computer Science & Engineering

UNIVERSITY of WASHINGTON



News & Events	People	Education	Research	Current Students	Prospective Students	Faculty Candidates	Alumni	Industry Affiliates	Support CSE
---------------	--------	-----------	----------	------------------	----------------------	--------------------	--------	---------------------	-------------

CSE 333 13su Exercise 12

out: Wednesday, July 31, 2013
due: Friday, August 2, 2013 by **9:00 am**.

Consider the following C++ file, named [ex12.cc](#): (right-click the link to download)

```
1  /*
2   * Copyright 2012 Steven Gribble
3   *
4   * This file is part of the UW CSE 333 course project exercises.
5   * (333exer).
6   *
7   * 333exer is free software: you can redistribute it and/or modify
8   * it under the terms of the GNU General Public License as published by
9   * the Free Software Foundation, either version 3 of the License, or
10  * (at your option) any later version.
11  *
12  * 333exer is distributed in the hope that it will be useful,
13  * but WITHOUT ANY WARRANTY; without even the implied warranty of
14  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
15  * GNU General Public License for more details.
16  *
17  * You should have received a copy of the GNU General Public License
18  * along with 333proj. If not, see <http://www.gnu.org/licenses/>.
19  */
20
21 #include <algorithm>
22 #include <cstdlib>
23 #include <iostream>
24 #include <vector>
25
26 int main(int argc, char **argv) {
27     // Allocates an integer on the heap, initializes to value 5.
28     int *x = new int(5);
29     std::cout << "x is: " << *x << std::endl;
30
31     // Allocates a vector of integers on the heap, adds some values to
32     // that vector, sorts the vector, prints out the values.
33     std::vector<int> *v = new std::vector<int>;
34     v->push_back(5);
35     v->push_back(9);
36     v->push_back(7);
37     std::sort(v->begin(), v->end());
38     std::cout << "sorted v: ";
39     for (int &el : *v) {
40         std::cout << el << " ";
41     }
42     std::cout << std::endl;
43
44     // Allocates a vector of (integer pointers) on the stack, adds some
45     // values to the vector from the heap, prints out the values.
46     std::vector<int*> v2;
47     v2.push_back(new int(5));
48     v2.push_back(new int(9));
49     v2.push_back(new int(7));
50     std::cout << "unsorted v2: ";
51     for (int *el : v2) {
52         std::cout << *el << " ";
53     }
54     std::cout << std::endl;
55
56     return EXIT_SUCCESS;
57 }
```

If you try compiling and running this program under valgrind, you'll notice that there are several memory leaks in it, all arising from allocating values on the heap using "new" but not deallocating them. Your challenge is to fix these memory leaks.

However, to make it interesting, you are not allowed to directly use "delete" (or "free", which would be a mistake, of course). Instead, you have to modify the code to use the std::unique_ptr smart pointer. So, for example, line 28 of the program allocates an integer on the heap using new. You will need to modify that line to transfer ownership of the allocated integer to a std::unique_ptr<int>.

The only modifications you are allowed to make to the code are those involved with adding in the appropriate std::unique_ptr support. When finished with your modifications, your code must compile without errors or warnings, and have no crashes, memory leaks, or memory errors.

You should compile your code with:

```
g++ -Wall -g -std=gnu++11 -o ex12 ex12.cc
```

Modify the comment at the top of your ex12.cc file with your name, student number, and CSE or UW email address, and submit your exercise

using the assignment dropbox linked on the main course web page.

Computer Science & Engineering University of Washington Box 352350 Seattle, WA 98195-2350 (206) 543-1695 voice, (206) 543-2969 FAX

[UW Privacy Policy](#) and [UW Site Use Agreement](#)