1. What header file must you #include to define vector objects?

<vector>

1. Write a definition statement for an empty vector of ints named scores.

Std::vector<int> scores;

1. Write a definition statement for a vector of 40 strings named student\_names.

Std::vector<std::string> student\_names(40);

1. Write a definition statement for a vector of 200 doubles named step\_increase, with each element initialized to 2.5.

Std::vector<double> step\_increase(200,2.5);

1. temperatures is an empty vector of floats. Write a statement that stores the value 2.7 in temperatures.

Temperatures.push\_back(2.7)

1. rainfall is a vector of doubles, with 10 elements. Write a statement that stores the value 12.897 in element 4 of the rainfall vector.

Rainfall[4] = 12.897;

1. Suppose we have the declaration

vector<int> v(10)

v.push\_back(9);

v.push\_back(8);

v.push\_back(7);

v.pop\_back();

v.push\_back(6);

Display the contents of the vector v.

0 0 0 0 0 0 0 0 0 0 9 8 6

1. To store a value in a vector that does not have a starting size, or that is already full, use the \_\_\_\_\_\_\_push\_back\_\_\_\_\_\_\_\_\_ member function.
2. Use the \_\_\_\_\_\_\_pop\_back\_\_\_\_\_\_\_\_\_ member function to remove the last element from a vector.
3. Write a statement to completely clear the contents of the student\_names vector from above.

Student\_names.clear();

1. Define a vector of integers that contains the first five prime numbers (2, 3, 5, 7, and 11), without using push\_back.

Std::vector<int> integers({2,3,5,7,11})

1. Write a function that takes in two arguments, lower\_bound, and upper\_bound. This function returns a vector containing all of the prime numbers between lower\_bound and upper\_bound.

vector<int> primeNums(int lower\_bound, int upper\_bound){

vector<int> primeNums(int lower\_bound, int upper\_bound){

vector<int> primeNums;

for(int i = lower\_bound; i <= upper\_bound; i++){

if(i%i == 0)

primeNums.push\_back(i);

}

return primeNums;

}

1. What is the contents of the vector names after the following statements?  
   vector<string> names;  
   names.push\_back("Ann");  
   names.push\_back("Bob");  
   names.pop\_back();  
   names.push\_back("Cal");

names.at(1) = "Bill";

names.push\_back("Bill");

names.at(2) = "Sam";

Ann, Bill, Sam

1. Suppose you want to store a set of temperature measurements that is taken every five minutes. Should you use a vector or an array?

Vector

1. Suppose you want to store the names of the weekdays. Should you use a vector or an array of seven strings?

Array of seven strings

1. Write code to perform the following tasks with vectors in C++?  
   a. Test that two vectors contain the same elements in the same order.

vector<int> v(**{**1,2**}**);  
vector<int> vv(**{**1,2**}**);  
vector<int> vvv(**{**3,2**}**);  
cout << (v == vv);  
cout << (v == vvv);

b. Copy one vector to another.

vector<int> v(**{**1,2**}**);  
vector<int> vvv(**{**3,2**}**);  
v = vvv;  
cout << v[0] << v[1];

c. Fill a vector with zeroes, overwriting all elements in it.

vector<int> v(**{**1,2**}**);  
  
v = vector<int>( v.size(),0);  
cout << v[0] << v[1];

d. Remove all elements from a vector.

v.clear();

1. Write a function  
   vector<int> append(vector<int> a, vector<int> b)  
   that appends one vector after another. For example, if a is  
   1 4 9 16  
   and b is  
   9 7 4 9 11  
   then append returns the vector  
   1 4 9 16 9 7 4 9 11

vector<int> append(vector<int> a, vector<int> b){

for(int i = 0; i <b.size();i++){

a.push\_back(b[i]);

}

return a;

}

1. Write a function  
   vector<int> merge(vector<int> a, vector<int> b)  
   that merges two vectors resulting in a vector that contains all of the elements sorted in order from smallest to largest. If one vector is shorter than the other, then alternate as long as you can and then append the remaining elements from the longer vector. You can assume the two vectors are sorted. For example, if a is  
   1 4 9 16  
   and b is  
   4 7 9 9 11  
   then merge returns the vector  
   1 4 4 7 9 9 9 11 16

vector<int> merge(vector<int> a, vector<int> b){

for(int i = 0; i <b.size();i++){

a.push\_back(b[i]);

}

std::sort(a.begin(), a.end());

return a;

}

1. What is the difference between the size and capacity of a vector?

Size is how many elements there are in the vector,

Capacity is how many elements the vector is allocated for.