Activity #10: Vectors
Recorder's Report

Manager: Reader:

Recorder: Driver:

Date: Score: Satisfactory / Not Satisfactory

Record your team's answers to the key questions (marked with ) below.

a) Model 1, Question #9

b) Model 2, Question #11 (final vector and accumulator only)

c) Model 3, Question #16.a

Activity #10: Vectors

In this course, you will work in teams of 3–4 students to learn new concepts. This activity will introduce you to vectors in C++.

Content Learning Objectives

After completing this activity, students should be able to:

- Read and write nested `for` loops.
- Identify inner and outer loops.

Process Skill Goals

During the activity, students should make progress toward:

- Write code that uses nested `for` loops.

Model 1 Vector Syntax

```
1  vector<int> quizScores = {8,6};
2  vector<string> profNames(4);
3
4  profNames.at(0) = "Carman";
5  profNames.at(1) = "Foster";
6  profNames.at(2) = "Duncan";
7
8  cout << quizScores.size() << endl;
9  cout << profNames.at(1) << endl;
10
```

quizScores:	8	6
index	0	1

profNames:	Carman	Foster	Duncan	
index	0	1	2	3

Output:

2
Foster

Refer to Model 1 above as your team develops consensus answers to the questions below.

Questions (15 min) Start time:

1. A *vector* is an ordered list of related variables, all of the same type. Each value in a vector is known as an *element*. The code above can be found in `activity10a.cpp`. Answer the following questions about the vectors declared on line 1.

a) What is the name of this vector?

b) What type of variable does this vector hold?

2. A second vector is declared on line 2 in the model. Answer the following questions about it.

a) What is the name of this vector?

b) What type of variable does this vector hold?

3. The *size* of a vector is the number of variables that it holds. What is the size of each vectors above?

4. Based on the code above, what are two different ways to initialize the variables stored in a vector?

5. The *index* of an element in a vector is its position in the vector. Indexes start at 0 in C++. Answer the following questions related to indexes.

a) What is stored at index 0 in the vector `quizScores`?

b) At what index is the name "Duncan" stored in the vector `profNames`?

c) What is the maximum allowed index for the vector `profNames`?

d) What is stored at index 2 in the vector `quizScores`?

6. Write a single line of C++ code to declare a new vector of doubles named `homeworkAvg` that contains the values 82.4 at index 2, 91.6 and index 0, and 73.9 at index 1.

7. The expression `myVector.at(i)` returns the value of the variable at index `i` in the vector `myVector`. It can also be used to set the value at that index, as seen in lines 9-11 of the model.

a) Write a single C++ expression for the average of the two values in `quizScores`.

b) Write a sequence of C++ commands that changes the contents of the vector `profNames` to those shown below without using the string literals `"Foster"` or `"Carman"`.

`profNames:`	Aamodt	Carman	Duncan	Foster
Index	0	1	2	3

8. The expression `myVector.size()` returns the size of the vector `myVector`. Use this function to write a `cout` command for each line of output below.

a) `There are 4 CS professors at WWU.`

b) `Total number of quiz points: 20` (assume 10 points per quiz)

9. True/False: `myVector.at(myVector.size())` returns the last value in the vector. 

True False

Model 2 Two C++ Code Snippets

```
1  // put first 100 powers of two in x
2  vector<double> x(100);
3  for (int i = 0; i < x.size(); i++) {
4    x.at(i) = pow(2,i);
5  }
6
```

```
1  // sum the elements of y
2  int sum = 0;
3  for (int i = 0; i < y.size(); i++) {
4    sum += y.at(i);
5  }
6
```

Refer to Model 2 above as your team develops consensus answers to the questions below.

Questions (15 min) Start time:

10. One of the biggest advantages of a vector is the ability to process them using loops. That is, to perform the same task for multiple elements. The file `activity10b.cpp` contains the loops from the model above.

a) Find the value of each expression involving `x` declared and initialized in the first code snippet.

(a) `x.at(1) + x.at(2)`

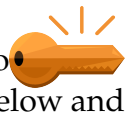
(b) `x.at(2) * x.at(3)`

(c) `x.at(4) / x.at(1)`

(d) `x.at(x.at(2))`

b) How many values are saved in the vector `y` in the second code snippet? Does it matter?

11. A *code trace* is a method for hand simulating the execution of your code in order to manually verify that it works before you compile it. Fill in the table to trace the code below and determine the value of the `data` vector and `accumulator` variable after the code has finished.



```
1  vecor<int> data = {5,26,13,12,37,15,16,4,1,3};
2  int accumulator = 0;
3  for (int i = 0; i < data.size(); i++) {
4    if (data.at(i) % 2 == 1 &&
5        i + 1 < data.size()) {
6      data.at(i) *= -1;
7      accumulator += data.at(i+1);
8    }
9  }
10
```

`data` =

`accumulator` =

i	data.at(i)	accumulator
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		

12. Suppose the vector `vector<double> a` and `vector<double> b` have been declared and filled with elements. Write code to find the pairwise maximum value in these vectors and place it in a vector named `myMax` which you declare. So, for example, `myMax.at(0)` should contain the larger of `a.at(0)` and `b.at(0)` and so on.

13. In a certain class 40% of your final grade comes from your homework average and 60% comes from your exam average. Suppose that vectors `vector<int> homeworkScores` and `vector<int> examScores` have been defined and contain your individual homework and exam scores. Write C++ code to compute your class grade and store it in `double finalGrade`. Don't forget to typecast if needed!

Model 3 Useful Vector Operations

Original `myVec`	Operation	Resulting `myVec`
myVec: | 8 | 2 | 6	`myVec.resize(4)`	myVec: | 8 | 2 | 6 |
myVec: | 8 | 2 | 6	`myVec.push_back(4)`	myVec: | 8 | 2 | 6 | 4
myVec: | 8 | 2 | 6	`myVec.pop_back()`	myVec: | 8 | 2
myVec: | 8 | 2 | 6	`myVec.insert(myVec.begin()+1,9)`	myVec: | 8 | 9 | 2 | 6
myVec: | 8 | 2 | 6	`myVec.erase(myVec.end()-2)`	myVec: | 8 | 6
myVec: | 8 | 2 | 6	`myVec.clear()`	myVec:

Refer to Model 3 above as your team develops consensus answers to the questions below.

Questions (20 min) Start time:

14. In your own words, describe what each of the following vector operations does.

a) `myVec.resize(n)`

b) `myVec.push_back(value)`

c) `myVec.pop_back()`

d) `myVec.begin()`

e) `myVec.insert(location,value)`

f) `myVec.end()`

g) `myVec.erase(location)`

h) `myVec.clear()`

15. Suppose the vector `myVec` is defined as shown below. Sketch a diagram of the contents of the vector after each set of vector operations. Start over with the original `myVec` for each part of this problem. This code can be found in `activity10c.cpp`.

`myVec:`	25	12	73	19	42

a) First set of commands

```
1   myVec.pop_back();
2   myVec.insert(myVec.begin()+2,55);
3   myVec.erase(myVec.begin()+1);
4
```

b) Second set of commands

```
1   myVec.resize(7);
2   myVec.at(5) = 31;
3   myVec.insert(myVec.end()-3,90);
4   myVec.push_back(27);
5
```

16. Suppose the vector `myVec` is defined as shown below. Give a sequence of commands that would produce the given vectors below.

myVec:	x	A	c	Y	w

a)

myVec:	A	h	c	n	w	V

b)

myVec:	x	A	w	Z

c)

myVec:	F	x	c	Y		w		