Activity #17: Reading from Files
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 #8

b) Model 2, Question #14

c) Model 3, Question #19

Activity #17: Reading from Files

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

Content Learning Objectives

After completing this activity, students should be able to:

- Explain how to open a text file for reading
- Explain how to handle errors with input streams
- Explain how to redirect a file from the command line to use as input

Process Skill Goals

During the activity, students should make progress toward:

- Write code that opens, reads from, and closes a file
- Write code that correctly handles input stream errors
- Write code that redirects a file from the command line to use as input

Model 1 A C++ Program and Input File

C++ Code

```
1  #include <iostream>
2  #include <fstream>
3  #include <string>
4
5  using namespace std;
6
7  int main() {
8    string sport;
9    ifstream fin;
10   int count = 1;
11
12   fin.open("sports.txt");
13   if (!fin.is_open()) {
14     cout << "Could not open file" << endl;
15     return 1;
16   }
17
18   while (fin >> sport) {
19     sport.at(0) = toupper(sport.at(0));
20     cout << "Sport " << count
21          << ": " << sport << endl;
22     count++;
23   }
24   fin.close();
25 }
26
```

sports.txt File

```
basketball
baseball
football
volleyball
tennis
golf
lacrosse
soccer
badminton
bowling
fortnight
skiing
diving
ice hockey
biking
rugby
swimming
sailing
rowing
skateboarding
```

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

Questions (15 min) Start time:

1. This code can be found in `activity17a.cpp`. Run it and then describe what the program does.

2. On line 12 of the C++ code, what does the string `"sports.txt"` represent?

3. Without discussing its relative merits as a sport, remove line 11 from the text file `"sports.txt"`, save your change, and rerun the program. What happened?

4. C++ can get input from a text file in much the same way it does from the keyboard. To do this it uses *file input streams*. Answer the questions below about the use of file input streams in this model.

a) What is the name of the library that includes these streams?

b) What is the type of variable we use for a file input stream?

c) The file input stream `fin` is used on line 12. What does this line do?

d) The file input stream `fin` is used again on line 13. What does this line do?

e) On what line does the program actually read input from the file?

f) On what line is the file closed (indicating we are done reading)?

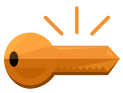
5. How is the sport named on line 14 of the text file printed by this program?

6. Below is a slight modification of this program (only lines 18-23 are shown). Make this modification in the file `activity17a.cpp` and record how the output changed.

```
1 while (getline(fin,sport)) {
2   sport.at(0) = toupper(sport.at(0));
3   cout << "Sport " << count
4        << ": " << sport << endl;
5   count++;
6 }
7
```

7. What is the difference between the command `fin >> sport` and the command `getline(fin, sport)`?

8. Rewrite the program so that it opens a user-entered filename and prints out the sports found inside. Test it with the file `"alternatives.txt"`.



Model 2 An Infinite Loop

```
1  #include <iostream>
2  #include <fstream>
3
4  using namespace std;
5
6  int main() {
7    ifstream fin;
8    int number;
9
10   fin.open("numbers.txt");
11   do {
12     fin >> number;
13     cout << "Number: " << number << endl;
14   } while (number != 6 && ! fin.eof());
15   fin.close();
16 }
17
```

numbers.txt File

```
2
four
6
```

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

Questions (20 min) Start time:

9. This code can be found in `activity17b.cpp`. What happens when you run it? Hint: To stop a program from executing, click on the terminal and press `Ctrl+C`.

10. What change to `"numbers.txt"` causes the program to print out 2, 4, and 6 and then exit?

11. For a file input stream `fin`, the function `fin.eof()` returns true if we've reached the *end-of-file* and false otherwise. Answer these questions to help you determine why the program runs in an infinite loop.

a) What does the code on line 12 do?

b) What type is the variable `number`?

c) What type of data is the `four` in the file?

d) When will the loop end?

e) If C++ tries to put something from an input stream into a variable and fails, the data being read is left on the input stream for later use. Why do you think the program goes into an infinite loop?

12. Modify the code in `activity17b.cpp` so that it reads a list of numbers from the keyboard and prints them out until the number 6 is entered. Record the changes you make.

13. Now test your program by entering `2`, `four`, and `6` from the keyboard. What happens?

14. The code below should be similar to what you wrote in problem 12, with some additions.

```
1  do {
2    cin >> number;
3    if (cin.fail()) {
4      cin.clear();
5      cin.ignore(100,'\n');
6    } else {
7      cout << "Number: " << number << endl;
8    }
9  } while (number != 6);
10
```

a) Update your code and run it again with inputs `2`, `four`, and `6`. What happens?

b) What do you think `cin.fail()` does?

15. Write a C++ program to print the sum of the numbers read from the text file `numbersTwo.txt` shown below. Any non-numeric data in the text file (such as the "`four`" in our model) should be ignored.

`numbersTwo.txt`

```
5
seven
3
12
15
ten
1
6
```

Model 3 Using a File for Keyboard Input

```cpp
#include <iostream>
#include <string>

using namespace std;

int main() {
  int number;
  string name;
  cout << "Enter Number: ";
  cin >> number;
  cout << "Enter Name: ";
  cin >> name;
  for (int i = 0; i < number; i++) {
    cout << name << endl;
  }
}

```

input.txt

```
5
Duncan
```

terminal commands

```
g++ activity17c.cpp -o test.o
./test.o < input.txt
```

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

Questions (15 min) Start time:

16. The file `activity17c.cpp` contains the code shown on the left-hand side of the model. Run this program as you normally would and describe what it does.

17. Now open a terminal in this same folder (right click on the folder name and pick *Open in Terminal*) and type in the commands shown in the model, pressing `Enter` after each line. What happens?

18. Modify the text file `input.txt` so that when you run the second terminal command in the model it prints out the name of one of your group members 15 times. What did your text file look like?

19. When you run a command `program < textfile` in the terminal, the contents of the text file are *redirected* to the *standard input* when the program is run. That is, the computer behaves as if you had typed the contents of the text file in at the keyboard while the program was running.

a) How does the program output differ when the input is redirected from a file instead of typed in?

b) Have you seen this type of output before?

20. Pick a previous homework assignment that requires user input. Write a text file with example input, compile the assignment, and run it with input redirected from your file. Record the assignment you chose and the input file you created below.