CMSC 202 Fall 2018

Project 1 – State Internet Access

**Assignment:** Project 1 – State Internet Access

**Due Date:** Thursday, September 27th at 8:59pm

**Value:** 80 points

# Overview

In this project, you will:

* Practice basic C++ syntax including branching structures
* Write a program that calls multiple functions
* Manage a two-dimensional array
* Use simple file input/output

# Background

Computer access and usage has grown considerably since the early 1980s. Only 8 percent of households in 1984 reported owning a computer according to a census of that year. Most adults were just learning to use computers then and they were focused on word processing, video games, and jobs. Just thirty years later, the percentage households with a computer had increased almost tenfold to 79 percent. By 2016, a new study found that 89 percent of households had a computer, making it a common feature of everyday life. Nowadays, people use computers for an even wider range of uses including online banking, entertainment, socializing, and accessing health care.

As part of our US census report, a report titled “Computer and Internet Use in the United States: 2016” looks at access to computers and broadband internet subscriptions nationwide. Differences can be seen by demographic, social and geographic characteristics across the digital divide between those who have and those who lack access to computers and the internet.

For this project, we are going to load data files with data about each of the states (including the District of Columbia) and display data related to that state. Additionally, we will then be using that data to create a simple bar graph displaying the percentages of those values.

The data from the source is included in Table 1. Internet Usage for Individuals 3 and older (2012). You do not need to use this table directly, a data file has been provided to you for the project.

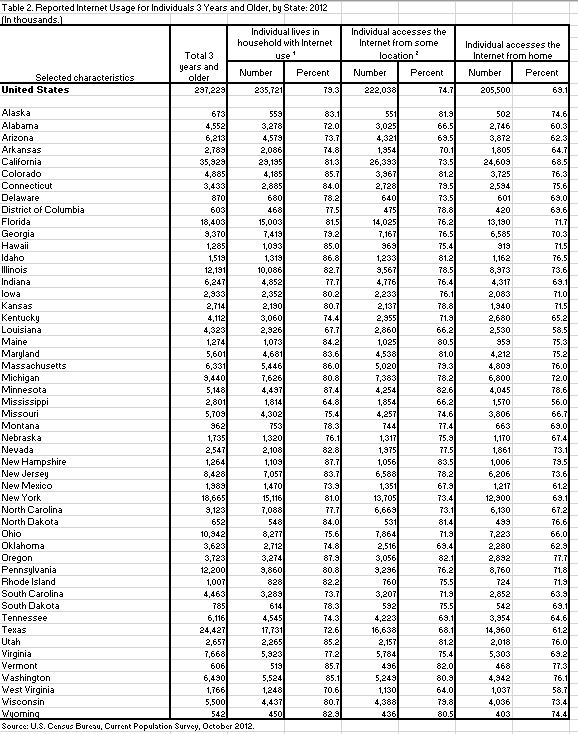


Table 1. Internet Usage for Individuals 3 and older (2012).

# Assignment Description

Your assignment is to develop a system to load in two specific demographic data files. The first data file will be used to store the states. The second data file will hold the demographic data. Once those files have been loaded, the data can be displayed and ultimately graphed.

The provided grid of data will be 51 x 4 (you will be expected to calculate the percentages which are not provided).

We will be using a simple column chart to visual the data. The columns should be: Percent of State Population (should always be 100%), Percent of State Population with Individuals that Live in Household with Internet Use, Percent of State Population with Individuals that Access the Internet from Some Location, and Percent of State Population with Individuals that Access the Internet from Home. The data files provided give populations and the actual percentages will need to be calculated. Figure 1 below shows the general format of the chart to be displayed (although for our project, we will use all text).

Figure 1. Maryland State Data

The equivalent for our project will look like figure 2 below:

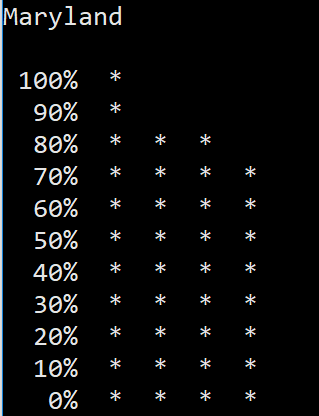


Figure 2. Maryland State Data

As you can see for our project, figure 2 has no labels or legend.

# Requirements:

This is a list of the requirements of this application. For this project, it is up to you exactly how you want to implement it. For you to earn all the points, however, you will need to meet all the defined requirements.

* You must follow the coding standard as defined in the CMSC 202 coding standards (found on Blackboard under course materials). This includes comments as required.
* The project must be turned in on time by the deadline listed above.
* The project must be completed in C++. You may not use any libraries or data structures that we have not learned in class. Libraries we have learned include **<iostream>, <fstream>, <iomanip>, <cmath>, <cstlib>,** and **<string>**. You should only use **namespace std**.
* You must use a variety of functions (at least 5) including passing parameters to those functions and returning information from those functions. At least one time, an array must be passed to a function (although you may do this more than once).
* The menu must be implemented with a switch statement.
* All user input must be validated. For example, if a menu allows for 1, 2, or 3 to be entered and the user enters a 4, it will re-prompt the user. However, the user is expected to always enter the correct data type. i.e. If the user is asked to enter an integer, they will. If they are asked to enter a character, they will. You do not need to worry about checking for correct data types.
* The name of the input file is variable. As a hint, don’t forget that the “open” command requires a c-string. If you want to convert a string to a c-string, you can use the string.c\_str() function.
* The easiest way to implement this project is using two-dimensional arrays (of the same size but of different data types).
* Have a main menu that asks if the user wants to:
  + Read in the demographic data files and display them in a formatted way.
  + Display the data for a specific state in a simple bar graph (graph labels are not required).
  + Exit and include a thank you message for the user
* Specific coding requirements include:
  + Must use at least 5 different functions.
  + Must use at least two different arrays.
  + Must pass an array to a function.
  + Must read in a file into a two-dimensional array.
  + Must use at least one switch statement.
  + Must use input validation (assume the data is the correct type).
  + Must use at least one do..while loop.
  + Must use constants as needed.
  + Must use setw to output the tables and graph.

# Recommendations

You are free to implement this with your own functions. While not required, these are some functions that you may want to include:

* + Main Menu – Welcomes the user to the application and has the user choose between loading demographic data, graphing demographic data, and exiting.
  + Load and Print Internet Data by State – Prompts the user for the name of the input files then displays the data loaded.
  + Graph Data by state – Prompts the user for which state (by number) then displays data about the state.
  + Display States – for use in the graph data function – lists each state with a number

# Sample Input and Output

For this project, input files are very simple. The first data file is called proj1\_states.txt and just includes the state names.

|  |
| --- |
| Alaska  Alabama  Arizona  Arkansas  California  Colorado  …  New Jersey  New Mexico  New York  North Carolina  North Dakota  Ohio  Oklahoma  Oregon  Pennsylvania  Rhode Island  South Carolina  South Dakota  Tennessee  Texas  Utah  Virginia  Vermont  Washington  West Virginia  Wisconsin |

The second file is all of the demographic data by state. The data is numbered in alphabetical order with the exception of Alaska and Alabama (for some reason the demographic data for Alaska and Alabama is in reverse order). So, 0 is Alaska and 1 in Alabama.

|  |
| --- |
| 0 673 559 551 502  1 4552 3278 3025 2746  2 6213 4579 4321 3872  3 2789 2086 1954 1805  4 35929 29195 26393 24609  5 4885 4185 3967 3725  6 3433 2885 2728 2594  7 870 680 640 601  8 603 468 475 420  9 18403 15003 14025 13190  10 9370 7419 7167 6585  11 1285 1093 969 919  12 1519 1319 1233 1162  13 12191 10086 9567 8973  14 6247 4852 4776 4317  15 2933 2352 2233 2083  16 2714 2190 2137 1940  17 4112 3060 2955 2680  18 4323 2926 2860 2530  19 1274 1073 1025 959  20 5601 4681 4538 4212  21 6331 5446 5020 4809  22 9440 7626 7383 6800  23 5148 4497 4254 4045  24 2801 1814 1854 1570  25 5709 4302 4257 3806  26 962 753 744 663  27 1735 1320 1317 1170  28 2547 2108 1975 1861  29 1264 1109 1056 1006  30 8428 7057 6588 6206  31 1989 1470 1351 1217  32 18665 15116 13705 12900  33 9123 7088 6669 6130  34 652 548 531 499  35 10942 8277 7864 7223  36 3623 2712 2516 2280  37 3723 3274 3056 2892  38 12200 9860 9296 8760  39 1007 828 760 724  40 4463 3289 3207 2852  41 785 614 592 542  42 6116 4545 4223 3954  43 24427 17731 16638 14960  44 2657 2265 2157 2018  45 7668 5923 5784 5303  46 606 519 496 468  47 6490 5524 5249 4942  48 1766 1248 1130 1037  49 5500 4437 4388 4036  50 542 450 436 403 |

The two data files can be downloaded from Prof. Dixon’s data folder by navigating to **your** project 1 folder and typing the following command:

**cp /afs/umbc.edu/users/j/d/jdixon/pub/cs202/proj1/proj1\_\*.txt .**

After you copy the test land cover files, you can type “**cat proj1\_data.txt**” or “**cat proj1\_states.txt**” and it should show you the entire file.

In the sample output below, user input is colored blue for clarity. After compiling and running proj1, the output would look like this:

|  |
| --- |
| Thank you for Using the Internet Demographic Tool  [jdixon@linux2 proj1]$ ./proj1  Welcome to the Internet Demographic Tool  What would you like to do?:  1. Load and Print Internet Demographic Data by State  2. Graph Internet Demographic Data by State  3. Exit |

If the user would choose 1 then the output would look like this:

|  |
| --- |
| 1  What is the name of the state file to import?  proj1\_states.txt  State file successfully loaded  What is the name of the data file to import?  proj1\_data.txt  Demographic file successfully loaded  Alaska 673 559 551 502  Alabama 4552 3278 3025 2746  Arizona 6213 4579 4321 3872  Arkansas 2789 2086 1954 1805  California 35929 29195 26393 24609  Colorado 4885 4185 3967 3725  Connecticut 3433 2885 2728 2594  Delaware 870 680 640 601  District of Columbia 603 468 475 420  Florida 18403 15003 14025 13190  Georgia 9370 7419 7167 6585  Hawaii 1285 1093 969 919  Idaho 1519 1319 1233 1162  Illinois 12191 10086 9567 8973  Indiana 6247 4852 4776 4317  Iowa 2933 2352 2233 2083  Kansas 2714 2190 2137 1940  Kentucky 4112 3060 2955 2680  Louisiana 4323 2926 2860 2530  Maine 1274 1073 1025 959  Maryland 5601 4681 4538 4212  Massachusetts 6331 5446 5020 4809  Michigan 9440 7626 7383 6800  Minnesota 5148 4497 4254 4045  Mississippi 2801 1814 1854 1570  Missouri 5709 4302 4257 3806  Montana 962 753 744 663  Nebraska 1735 1320 1317 1170  Nevada 2547 2108 1975 1861  New Hampshire 1264 1109 1056 1006  New Jersey 8428 7057 6588 6206  New Mexico 1989 1470 1351 1217  New York 18665 15116 13705 12900  North Carolina 9123 7088 6669 6130  North Dakota 652 548 531 499  Ohio 10942 8277 7864 7223  Oklahoma 3623 2712 2516 2280  Oregon 3723 3274 3056 2892  Pennsylvania 12200 9860 9296 8760  Rhode Island 1007 828 760 724  South Carolina 4463 3289 3207 2852  South Dakota 785 614 592 542  Tennessee 6116 4545 4223 3954  Texas 24427 17731 16638 14960  Utah 2657 2265 2157 2018  Virginia 7668 5923 5784 5303  Vermont 606 519 496 468  Washington 6490 5524 5249 4942  West Virginia 1766 1248 1130 1037  Wisconsin 5500 4437 4388 4036  Wyoming 542 450 436 403  What would you like to do?:  1. Load and Print Internet Demographic Data by State  2. Graph Internet Demographic Data by State  3. Exit |

Here is another run where we do the first modeling.

|  |
| --- |
| 2  Which state would you like to graph?  1 Alaska  2 Alabama  3 Arizona  4 Arkansas  5 California  6 Colorado  7 Connecticut  8 Delaware  9 District of Columbia  10 Florida  11 Georgia  12 Hawaii  13 Idaho  14 Illinois  15 Indiana  16 Iowa  17 Kansas  18 Kentucky  19 Louisiana  20 Maine  21 Maryland  22 Massachusetts  23 Michigan  24 Minnesota  25 Mississippi  26 Missouri  27 Montana  28 Nebraska  29 Nevada  30 New Hampshire  31 New Jersey  32 New Mexico  33 New York  34 North Carolina  35 North Dakota  36 Ohio  37 Oklahoma  38 Oregon  39 Pennsylvania  40 Rhode Island  41 South Carolina  42 South Dakota  43 Tennessee  44 Texas  45 Utah  46 Virginia  47 Vermont  48 Washington  49 West Virginia  50 Wisconsin  51 Wyoming  21  Maryland  100% \*  90% \*  80% \* \* \*  70% \* \* \* \*  60% \* \* \* \*  50% \* \* \* \*  40% \* \* \* \*  30% \* \* \* \*  20% \* \* \* \*  10% \* \* \* \*  0% \* \* \* \*  What would you like to do?:  1. Load and Print Internet Demographic Data by State  2. Graph Internet Demographic Data by State  3. Exit |

Here are some example of input validation:

|  |
| --- |
| ./[jdixon@linux2 proj1]$ ./proj1  Welcome to the Internet Demographic Tool  What would you like to do?:  1. Load and Print Internet Demographic Data by State  2. Graph Internet Demographic Data by State  3. Exit  1  What is the name of the state file to import?  states.txt  The file states.txt was not opened  What is the name of the data file to import?  data.txt  The file data.txt was not opened  Data not successfully loaded  What would you like to do?:  1. Load and Print Internet Demographic Data by State  2. Graph Internet Demographic Data by State  3. Exit |

Here is another example where you try to graph before the data files are loaded:

|  |
| --- |
| ./[jdixon@linux2 proj1]$ ./proj1  Welcome to the Internet Demographic Tool  What would you like to do?:  1. Load and Print Internet Demographic Data by State  2. Graph Internet Demographic Data by State  3. Exit  2  State File not loaded  Choose load and print first  What would you like to do?:  1. Load and Print Internet Demographic Data by State  2. Graph Internet Demographic Data by State  3. Exit |

Finally, here is an example where we choose options that are outside of our expected range. There are several examples of this in our project:

|  |
| --- |
| What would you like to do?:  1. Load and Print Internet Demographic Data by State  2. Graph Internet Demographic Data by State  3. Exit  0  Please choose again  What would you like to do?:  1. Load and Print Internet Demographic Data by State  2. Graph Internet Demographic Data by State  3. Exit  4  Please choose again  What would you like to do?:  1. Load and Print Internet Demographic Data by State  2. Graph Internet Demographic Data by State  3. Exit |

# Compiling and Running

To compile your program, enter the following command at the Linux prompt:

g++ -Wall proj1.cpp -o proj1 (use this command to show warnings – which should be eliminated before turning your code in)

This command runs the GNU C++ compiler (**g++**). The option **-Wall** instructs the compiler to be verbose in its production of warning messages; the option **-o proj1** (hyphen followed by the letter "o", not the digit zero), instructs the compiler to give the executable program the name **proj1**. If the program compiles correctly, the executable file **proj1** will be created in the current directory. Your project files should have no warnings or errors when turned in.

At the Linux prompt, enter the command **./proj1** to run your program. It should look like the sample output provided above.

# Completing your Project

When you have completed your project, you can copy it into the submission folder. You can copy your files into the submission folder as many times as you like (before the due date). We will only grade what is in your submission folder.

For this project, you should submit these files to the **proj1** subdirectory:

**proj1.cpp** — should include your implementations of the required functions.

Submitting your project has two steps:

1. Set up a symbolic link in your home directory to your submission folder. Execute these commands:
   1. **cd ~**
   2. **ln -s /afs/umbc.edu/users/j/d/jdixon/pub/cmsc202/$USER ~/cs202proj**
   3. To check that the symbolic link was built successfully, you can type:
      1. **ls ~/cs202proj**
      2. **It should list proj1, proj1-late1, proj1-late2 through proj5-late2**
2. Copy the project files into your proj1 folder. Execute these commands:
   1. cd to your project 1 folder. An example might be:

**cd ~/202/projects/proj1**

* 1. **cp proj1.cpp ~/cs202proj/proj1**

You can check to make sure that your files were successfully copied over to the submission directory by entering the command

ls ~/cs202proj/proj1

You can check that your program compiles and runs in the **proj1** directory, but please clean up any **.o** and executable files. Again, do not develop your code in this directory and you should not have the only copy of your program here.

**IMPORTANT:** If you want to submit the project late (after the due date), you will need to copy your files to the appropriate late folder. If you can no longer copy the files into the proj1 folder, it is because the due date has passed. You should be able to see your proj1 files but you can no longer edit or copy the files in to your proj1 folder. (They will be read only)

* If it is 0-24 hours late, copy your files to **~/cs202proj/proj1-late1**
* If it is 24-48 hours late, copy your files to **~/cs202proj/proj1-late2**
* If it is after 48 hours late, it is too late to be submitted.