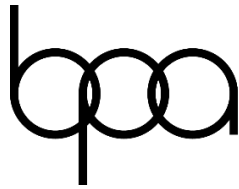


Contestant ID: _____

Time: _____

Rank: _____



**BUSINESS
PROFESSIONALS**
of AMERICA
Giving Purpose to Potential

C++ Programming (335)

REGIONAL 2026

PRODUCTION:

Regional_C++ Menu Project

_____ (540 points)

Test Time: 90 minutes

GENERAL GUIDELINES.

Failure to follow any of these rules may result in disqualification:

1. **Submission Requirements:** Contestants must submit this test booklet along with any printouts.
2. **Permitted Items:** Only the equipment, supplies, and materials specified for this event are allowed in the testing area. Previous BPA tests and sample tests (whether handwritten, photocopied, or typed) are not permitted.
3. **Electronic Devices:** Electronic devices will be monitored according to ACT standards.

EXAM GUIDELINES.

1. Your name and/or school name should *not* appear on work you submit for grading.
2. Create a folder on the flash drive provided using your contestant number as the name of the folder.
3. Copy your entire solution/project into this folder.
4. Submit your entire solution/project so that the graders may open your project to review the source code.
5. Ensure that the files required to run your program are present and will execute on the flash drive provided.

*Note that the flash drive letter may *not* be the same when the program is graded as it was when you created the program. Use relative file paths, not absolute file paths.

The graders will *not* alter your source code. Submissions that do *not* contain source code will *not* be graded. The project must include an executable program file.

Commenting for Source Code Review:

- Certain sections of your code will be graded. These gradable blocks of code can range from creating data structures, method algorithms, exception handling, and class construction.
 - **Code Commenting Requirements:** Clear and concise comments must be included for all major components of the program, specifically:
 - **Input:** Document how data is received or acquired, including user inputs, file reads, the format of the input, API requests, etc.
 - **Processing:** Provide comments that explain the logic, algorithms, or transformations applied to the input data.
 - **Output:** Describe how and where the final results are produced, the format of the output, whether through user interfaces, files, or other systems.
 - These comments should enhance understanding of the program's flow and intent, making it easier for others to read, maintain, and debug the code.
- The grading rubric contains a section called Source Code Review: in this section are listed descriptions of all the graded programming concepts.
- Each gradable item and any significant programming area should be commented. There will be points awarded for proper commenting.

C++ Regional

You will be developing a console application with C++. The goal of this application is to create a simple menu system that will perform various actions to demonstrate your programming skills at the regional level competition. Your program will need to capture all the important data from the user and/or file as well as handle any exceptions or data entry errors.

Menu Options You Will Be Programming

When the program begins, you will see a menu appear in the console. There are 5 selections that the user will be able to select. Each menu item can be programmed independently, which means you can pick and choose which menu items you want to program. Below the image (Figure 1) is a description of each menu item. Your menu should handle any invalid inputs that are not option 1 to 5.

General Notes for All Menu Options:

1. The program should gracefully handle any invalid input by prompting the user to try again without crashing.
2. All numeric inputs should reject letters, symbols, decimals, and negative values where integers are expected.
3. User inputs must be validated to ensure they fall within appropriate ranges or match required formats.
4. Each menu option should have its own `function/method.

```
===== Main Menu =====  
1. Read and display CSV file  
2. Generate random passwords  
3. Reverse a sentence  
4. Simple calculator  
5. Exit  
=====  
Please select an option (1-5): |
```

Figure 1

Menu Option 1: Read and Display CSV File

Task Description:

The student must write a function named “readCSV”. This function will take in no information and will return no information. It will read, parse and display the contents of the provided CSV file (**computer.csv**) in a structured, tabular format. The output should include column headers and rows of data. They need to ensure proper formatting based and handle any file-related errors, such as the file being missing or inaccessible.

Expected Behaviors:

- If the file is missing or unreadable, the program should display an appropriate error message.
- Data must be displayed in a neatly aligned table with headers as shown in figure 2.

```
Please select an option (1-5): 1
Reading file: computers.csv
=====
ComputerID    Brand        Processor    RAM    Storage
-----
1             Dell         Intel i5     8GB    256GB SSD
2             HP           Intel i7     16GB   512GB SSD
3             Lenovo       AMD Ryzen 5  8GB    1TB HDD
4             Apple        M1           16GB   512GB SSD
5             Acer         Intel i3     4GB    128GB SSD
6             Asus         Intel i9     32GB   1TB SSD
7             Microsoft   Intel i5     8GB    256GB SSD
8             Samsung     AMD Ryzen 7  16GB   1TB SSD
9             Toshiba     Intel i3     4GB    500GB HDD
10            Sony         Intel i7     16GB   1TB SSD
=====
Do you want to select another option? (y/n): |
```

Figure 2

Menu Option 2: Generate Random Passwords

Task Description:

The student needs to write a function named “randomPassword” to generate a user provided number of secure random passwords. The number must be between 1 and 9 inclusively. Each password should meet complexity requirements (e.g., include uppercase letters, lowercase letters, digits, and symbols).

Expected Behaviors:

- Prompt the user to re-enter valid input if the input is non-integer or out of range of 1 - 9.
- Generate passwords that meet the complexity criteria and display them. Each password needs to be 8 characters long. The password can consist of any of the following characters:
abcdefghijklmnopqrstuvwxyz1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ
WZ!@#\$%

```
Please select an option (1-5): 2
Enter the number of passwords to generate: 5
6cWU%QHi
Ag5otZg#
U03PGr#K
pV6xl$JC
tqUj9cWz
=====
Do you want to select another option? (y/n): |
```

Menu Option 3: Reverse a Sentence

Task Description:

The student must write a function named “reverseSentence”. This function will take in no information and will return no information. It will prompt the user to input a sentence, reverse the characters in the sentence, and display the reversed string. Punctuation and spaces must also be reversed. See figure 3.

Expected Behaviors:

- If the user enters no input, prompt them to enter a valid sentence.
- Display the reversed sentence, including spaces and punctuation.

```
Please select an option (1-5): 3
Enter a sentence: This is the sentence that I want to reverse.
Reversed sentence: .esrever ot tnaw I taht ecnetnes eht si sihT
=====
Do you want to select another option? (y/n): |
```

Menu Option 4: Simple Calculator

Task Description:

The student will write a function named “simpleCalculator”. This function will take in no information and will return no information. This function will implement a simple calculator that performs addition, subtraction, multiplication, or division operations based on user input. The programmer can assume that all numeric data has a floating point. See figure 4.

Improper Input:

- Invalid operators (anything other than +, -, *, /).
- Non-numeric values for the operands.
- Division by zero.

Expected Behaviors:

- Prompt the user to re-enter valid input for invalid operators or non-numeric operands.
- For division display an error if the user attempts to divide by zero and return to the main menu

```
Please select an option (1-5): 4
Enter the first number: 11
Enter the operator (+, -, *, /): /
Enter the second number: 3
Result: 3.66667
=====
Do you want to select another option? (y/n): |
```

Menu Option 5: Exit

Task Description:

The student will write a function named “exitProgram”. This function will take in no information, but it will return a boolean value on whether or not to exit the program. This function will be call from the menu after each iteration of the menu.

Expected Behaviors:

- Closes the program or returns the menu.

Requirements/Explanations:

1. You must create a C++file called: **Regional_MenuProject.cpp**
2. The computers.csv file has been given to you; you are responsible for placing it in a location that allows the program to import the data from a relative location so that your program files can be uploaded into the grader computer and operate as intended.

Rubric:

Solution and Project		
The C++ project file is correctly named and present on the flash drive in a single folder with your contest ID		10
Program Execution (<i>If the program does not execute, then the remaining items in this section receive a score of zero</i>). All menu options must work according to the instructions.		
Main Menu (appearance and performance)		20
#1 Read and Display CSV File in tabular form that is evenly spaced and aligned correctly		40
#1 Invalid Input for Read and Display CSV File		10
#2 Generate Random Passwords		50
#2 Invalid Input for Generate Random Passwords		20
#3 Reverse a Sentence		30
#3 Invalid Input for Reverse a Sentence		10
#4 Simple Calculator		30
#4 Invalid Input for Simple Calculator		20
#5 Exit		20
Subtotal		/260 Points

Source Code Review		
Program is properly commented (See Commenting for Source Code Review above)		50
Main menu is validated and handles all user inputs		40
#1 Read and Display the contents CSV File: Reads and parsed all data in the CSV file. Outputs the values in tabular for as shown in the sample provided		50
#2 Generate Random Passwords: Creates the number of passwords requested by the user. Uses characters provided in the instructions.		50
#3 Reverse a Sentence: Reverses the string entered by the user. Checks for an empty string.		30
#4 Simple Calculator: Validates user input. Calculates the answer based on user input		50
#5 Exit: Asks the user if they would like to select another option. Returns a Boolean value.		10
Subtotal		/280 Points
Total Points		/540 Points