



C++ PROGRAMMING (335)

REGIONAL – 2014

TOTAL POINTS

_____ (*300 points*)

Judges/Graders: Please double check and verify all scores and answer keys!

Property of Business Professionals of America.
May be reproduced only for use in the Business Professionals of America
Workplace Skills Assessment Program competition.



Technical Task Scoring Sheet:

Application/Execution

- Application presents user with correct options menu _____ 10 pts
- Application “Display inventory” menu item correctly displays all items from “gameinventory.txt” _____ 30 pts
- Application “Add Item” menu item correctly prompts user for new item attributes and correctly appends the item to the file _____ 40 pts
- Application “Quit” menu item successfully exits program _____ 10 pts
- Screen is cleared after each menu operation _____ 10 pts

Application/Code Structure

- File input instance variables are declared and initialized _____ 20 pts
- File output instance variables are declared and initialized _____ 20 pts
- I/O error handling is done if files cannot be opened _____ 20 pts
- Method to display inventory items is implemented (display_inventory) _____ 40 pts
- Method to add new inventory item is implemented (add_item) _____ 40 pts
- Invalid menu choice error handling implemented _____ 20 pts
- Methods are properly commented _____ 10 pts
- Variables are properly commented _____ 10 pts
- Code copied to USB drive and program runs from USB _____ 20 pts

Total Points: _____ 300 pts

Note to Graders:

In order to properly test file error handling, remove the file (gameinventory.txt) from the flash drive after you have tested the other program execution rubric items and see if the contestant properly warns the user of a “file not found exception”.



Sample Solution code. Contestant code may vary.

```
//CONTESTANT ID #
//PROGRAM DESCRIPTION
#include <iostream>
#include <string>
#include <fstream>
#include <cstdlib>

using namespace std;

// Method Section
void display_inventory();
void add_item();

// Display Item Inventory
void display_inventory()
{
    //Create character array for file reading input
    char input[80];

    //Create an input file object
    ifstream inFile;

    inFile.open("gameinventory.txt"); // Open the file

    if (inFile.fail()) //File Exception Handling
    {
        cout << "File not found!, exiting" << endl ;
        system("PAUSE");
        exit(1);
    }

    while (!inFile.fail()) //Read data until file input fails
    {
        inFile.getline(input, 80); //Read one line at a time
        cout << input<< endl; //Output the read line to the user
    }
}
```



```
inFile.close();           //Close the file

cout << "Press a key to return to main menu."; //Wait for user input
cin.ignore();
cin.get();

}

// Append a new item to the end of the file
void add_item()
{
    string name;    //New Item Name
    int cost;      //New Item Cost
    int quantity;  //New Item Quantity

    //Prompt user for new item attributes
    cout << "Enter the name for this new item: ";
    cin >> name;
    cin.ignore(80, '\n');
    cout << "Enter unit cost for this item: ";
    cin >> cost;
    cin.ignore(80, '\n');
    cout << "Enter quantity in stock for this item: ";
    cin >> quantity;
    cin.ignore(80, '\n');

    ofstream outFile;           //Create output file object

    outFile.open("gameinventory.txt", ios::app); //Open the output file,
                                                //include ios:app for file
                                                //appending

    //Commit data to the file
    outFile << name << "\t" << cost << "\t" << quantity << endl;

    outFile.close();           //close the file

    cout << "Item added to the inventory!" << endl;
    cout << "Press a key to return to main menu."; //Wait for user input
    cin.get();
}
```



```
}

int main()
{
    char choice;    //User menu choice
    int flag = 0;   //Flag used to check for program exit

    while (flag == 0)
    {
        //Display Menu Options
        cout << "\t\t\n" << "*****";
        cout << "\t\n" << "Game Item Inventory";
        cout << "\t\n" << "*****";
        cout << "\t\n" << "[d] Display all the inventory from file.";
        cout << "\t\n" << "[a] Append an item to the inventory file.";
        cout << "\t\n" << "[q] Quit and commit changes to file.";
        cout << "\t\n" << "Choice (enter upper or lowercase letter option): ";
        cin >>choice;

        //Check user menu choice
        switch(choice)
        {
            case 'd':
            case 'D':    //Display Inventory Method Call
                system("cls"); //Clear Screen
                display_inventory();
                system("cls"); //Clear Screen
                break;

            case 'a':
            case 'A':    //Add Item Method Call
                system("cls"); //Clear Screen
                add_item();
                system("cls"); //Clear Screen
                break;
        }
    }
}
```



```
case 'q':
case 'Q':      //Quit the program, prompt for assurance

    flag = 1;    //Set flag to exit condition
    break;

default:
    cout << "\nInvalid selection. Press a key to return to main menu.";
    cin.ignore();
    cin.get();
    system("cls");
}

if (flag == 1) //Exit flag has been set, break the loop
{
    break;
}

return 0;
}
```