Introduction to C++

|  |  |
| --- | --- |
| **Assessment Task Number:** Part 2 – Debugging Problems | |
| **Unit Code(s):** | **Unit Title(s):** |
| ICTPRG443 | Apply intermediate programming skills in different languages |
| ICTICT449 | Use version control systems in development environments |
| **Instructions to Learners:** | |

Download the student samples GitHub repository containing the project to debug from GitHub:

<https://github.com/AcademyOfInteractiveEntertainment/AIEYear1Samples>

You have received the following email from your supervisor:

Hi,

I have a problem with a bit of code one of the other interns has made.

The designers wanted a simple program to let them double-check the NPC data stored in the game’s data file.

In the *AIEYear1Samples* git repo you’ll find the project *IntroCPP\_BinaryFileIO\_NPCData*.

There are a few problems I have with this program:

* + - The designers tell me when they run this program on their real data files it takes way too long to load.   
      It seems the program reads the whole data file and stores the whole thing in memory – this is bad!  
      I need it to only load the current record. You’ll need to create a random-access algorithm to jump directly to the required record. DON’T sequentially read the file!
    - For some reason the NPC’s name isn’t being read in correctly. Can you fix this?
    - I get a crash when pressing the right arrow to go the next record while I’m on the last one (there are only 5 records in the sample file).
    - And finally, the most annoying thing is the whole program has no comments! Please add comments so the next intern knows what they’re doing.

When you’re done with your fixes, I want you to really *TEST* your work. Give me a brief report of how you tested your work – including what tests you ran and their results (I’m expecting at least 3 or 4 tests).

I also need to make sure you know what you’re doing in Visual Studio, so give me a few screenshots of you using the debugger. Show me:

* + - Some breakpoints you’ve placed in the code,
    - Some auto and custom watch variables displayed while debugging, and
    - The callstack while you’re debugging the program

Oh, and tell me the keyboard shortcuts for stepping through code in the debugger.

Do a good job and I’ll recommend you to the boss!

Cheers,  
Sam

Fix the problems described in the email above. Your final program must resolve all bugs, and use a random-access algorithm to read and load a single record at a time from the data file.

|  |  |  |
| --- | --- | --- |
| **Task** | | **Evidence Criteria** |
| 1. | Random Access File I/O | Update the given program to read a file using a random-access algorithm |
| 2. | Fix Name Display | Resolve the issue with the name displaying incorrectly |
| 3. | Program Crash | Resolve the issue with the program crashing when getting the next record |
| 4. | Comment Code | Add comments to the program.  Comments must be in line with industry standards or as defined by your trainer/assessor. |
| 5. | Testing Document | Document at least 3 test cases.  Documentation for each test case must list   * What is being tested * The input given * The expected output * The actual output * Any modifications made as a result of testing |
| 6. | Visual Studio Screenshots | A document containing screenshots of the following Visual Studio debugging features:   * Some breakpoints you’ve placed in the code, * Some auto and custom watch variables displayed while debugging, and * The callstack while you’re debugging the program |
| 7. | Debugging Shortcuts | A document containing a list of Visual Studio debugging shortcuts for navigating through code while using the debugger, including:   * Step Into * Step Over * Step Out |
| **Submission Requirements:** | | |
| You will need to submit the following:   * A Release build of each application that can execute as a stand-alone program * Your complete Visual Studio project   Be sure to remove any temporary build folders (i.e., the Debug and Release folders). Only project files, source code files, and any resource files used should be included in your submission.  Package all files in a single compressed archive file (.zip, .7z, or .rar) | | |

1. Random Access File I/O

Update the given program to read a file using a random-access algorithm

2. Fix Name Display

Resolve the issue with the name displaying incorrectly

3. Program Crash

Resolve the issue with the program crashing when getting the next record

4. Comment Code

Add comments to the program.

Comments must be in line with industry standards or as defined by your trainer/assessor.

5. Testing Document

Document at least 3 test cases.

Documentation for each test case must list

• What is being tested

• The input given

• The expected output

• The actual output

• Any modifications made as a result of testing

6. Visual Studio Screenshots

A document containing screenshots of the following Visual Studio debugging features:

• Some breakpoints you’ve placed in the code,

• Some auto and custom watch variables displayed while debugging, and

• The callstack while you’re debugging the program

7. Debugging Shortcuts

A document containing a list of Visual Studio debugging shortcuts for navigating through code while using the debugger, including:

• Step Into

• Step Over

• Step Out