

CSE 241 Programming Assignment 5

NOTE

The weight-point of this assignment is two times the weight-point of a regular programming assignment.

DUE

June 8th, 2021, 23:55

Description

- This is an individual assignment. Please do not collaborate
- If you think that this document does not clearly describes the assignment, ask questions before its too late.

This assignment is about using C++ STL, exception handling and/or creating Class templates.

- Your program reads two files:
 - `data.txt`
 - `commands.txt`
- According to content in `data.txt`, the program utilizes necessary STL classes and/or user-created classes for a catalog representation.
- Your program creates a log file(`output.txt`) for certain steps of operations performed on catalog.

`data.txt`

- This file holds information about a catalog. A catalog can be one of the following:
 - Book catalog
 - Music catalog
 - Movie catalog
- The type of the catalog is specified in the first line of `data.txt`

Book Catalog

- Each line in a book catalog keeps information about a book.
- Format:

`<title> <authors> <year> <tags>`

- Example: Contents of `data.txt` for a book catalog

```
1 book
2 "Hilbert Spaces With Applications" "Lokenath Debnath1, Piotr Mikusinski" "2005" "Mathematics,
  ↳ Set Theory"
3 "The Neolithic Revolution in the Near East: Transforming the Human Landscape" "Alan H. Simmons"
  ↳ "2011" "Social Science, Anthropology, Cultural, General, Archaeology"
4 "Learning Flask Framework" "Matt Copperwaite, Charles Leifer" "2015" "Computers, Programming
  ↳ Languages, Python, Internet, Application Development, Web, Web Programming"
5 "Graphics Gems V" "Alan W. Paeth" "1995" ""
```

Music Catalog

- Each line in a music catalog keeps information about a music album.
- Format:

`<title> <artists> <year> <genre>`

- Example: Contents of `data.txt` for a music catalog

```

1 music
2 "Professor Satchafunkilus and the Musterion of Rock" "Joe Satriani" "2008" "Guitar Virtuoso"
3 "Physical Graffiti" "Led Zeppelin" "1975" "Rock"
4 "Witchdoctor's Son" "Okay Temiz, Johnny Dyani" "2017" "Jazz, Fusion"
5 "Return Of The Mother Head's Family Reunion" "Richie Kotzen" "2007" "Rock, Guitar Virtuoso"

```

Movie Catalog

- Each line in a movie catalog keeps information about a movie.
- Format:

```
<title> <director> <year> <genre> <starring>
```

- Example: Contents of `data.txt` for a movie catalog

```

1 movie
2 "The Lord of the Rings: The Fellowship of the Ring" "Peter Jackson" "2001" "Adventure, Drama,
  ↳ Fantasy" "Elijah Wood, Ian McKellen, Orlando Bloom"
3 "Twelve Monkeys" "Terry Gilliam" "1995" "Mystery, Sci-Fi, Thriller" " Bruce Willis, Madeleine
  ↳ Stowe, Brad Pitt"
4 "Perfume: The Story of a Murderer" " Tom Tykwer" "2006" "Crime, Drama, Fantasy" "Ben Whishaw,
  ↳ Dustin Hoffman, Alan Rickman"
5 "Cold Mountain" "Anthony Minghella" "2003" "Adventure, Drama, History" " Jude Law, Nicole
  ↳ Kidman, Renee Zellweger"

```

commands.txt

This file includes several commands which work on the catalog information you read from `data.txt`. Each line is a command. The following should be recognized:

- There are two commands.

```
search <string> in <field>
sort <field>
```

search command

- Format:
`search <string> in <field>`
- Output:

This command returns a list of matched (partially or fully) entries (one entry in a line). Search should be limited to the field specified. Not found returns no line.

- Example:

```
search "Joe" in "artists"
```

This returns the following line:

```
"Professor Satchafunkilus and the Musterion of Rock" "Joe Satriani" "2008" "Guitar Virtuoso"
```

sort command

- Format:
`sort <field>`
- Output:

This command returns a list of sorted entries (ascending order)

- Example:

```
sort "title"
```

This returns the following lines:

```
1 "Cold Mountain" "Anthony Minghella" "2003" "Adventure, Drama, History" "Jude Law, Nicole
  ↳ Kidman, Renee Zellweger"
2 "Perfume: The Story of a Murderer" "Tom Tykwer" "2006" "Crime, Drama, Fantasy" "Ben Whishaw,
  ↳ Dustin Hoffman, Alan Rickman"
3 "The Lord of the Rings: The Fellowship of the Ring" "Peter Jackson" "2001" "Adventure, Drama,
  ↳ Fantasy" "Elijah Wood, Ian McKellen, Orlando Bloom"
4 "Twelve Monkeys" "Terry Gilliam" "1995" "Mystery, Sci-Fi, Thriller" "Bruce Willis, Madeleine
  ↳ Stowe, Brad Pitt"
```

output.txt

This file keeps the log of the operations. The following events should be recorded in the specified format:

- catalog read
- output of commands

catalog read

- First specify the type of the catalog.
- At the end, state the number of unique entries.

```
Catalog Read: music
4 unique entries
```

output of commands

- State the command.
- Append its output.

```
search "Joe" in "artists"
"Professor Satchafunkilus and the Musterion of Rock" "Joe Satriani" "2008" "Guitar Virtuoso"
```

Exceptions

- Your program should catch certain exceptions and create log entries for them.
- You need to catch the following exceptions:

Missing field exception

- If any of the field in any entries is missing your program should omit that line and create an exception record in the log file.

```
Exception: missing field
```

Duplicate entry exception

- If the first field of any entries fully match, your program should create an exception for each repetition and log these instances in the log file.

```
Exception: duplicate entry
```

Wrong command exception

- If the command is not in the expected format(unrecognized field name, extra information etc...), log this instance as an exception.

```
Exception: command is wrong
```

A full example.

- Suppose we are given the following `data.txt` file and `commands.txt` file:
- `data.txt`

```
1 movie
2 "The Lord of the Rings: The Fellowship of the Ring" "Peter Jackson" "2001" "Adventure, Drama,
  ↳ Fantasy" "Elijah Wood, Ian McKellen, Orlando Bloom"
3 "Twelve Monkeys" "Terry Gilliam" "1995" "Mystery, Sci-Fi, Thriller" " Bruce Willis, Madeleine
  ↳ Stowe, Brad Pitt"
4 "Twelve Monkeys" "" "" "Sci-Fi, Thriller" " Bruce Willis, Madeleine Stowe, Brad Pitt"
5 "Perfume: The Story of a Murderer" " Tom Tykwer" "2006" "Crime, Drama, Fantasy" "Ben Whishaw,
  ↳ Dustin Hoffman, Alan Rickman"
6 "Twelve Monkeys" "" "" "Mystery, Sci-Fi, Thriller" " Bruce Willis, Madeleine Stowe, Brad Pitt"
7 "Cold Mountain" "Anthony Minghella" "2003" "Adventure, Drama, History"
```

- `commands.txt`

```
1 search "Monkeys" in "title"
2 search "Spice" in "type"
3 sort "year"
```

- The first line is `movie`. This means your application will going to run in `movie organiser` mode.
- Following is the log file for this example:
- `output.txt`

```
1 Catalog Read: movie
2 Exception: duplicate entry
3 "Twelve Monkeys" "" "" "Sci-Fi, Thriller" " Bruce Willis, Madeleine Stowe, Brad Pitt"
4 Exception: duplicate entry
5 "Twelve Monkeys" "" "" "Mystery, Sci-Fi, Thriller" " Bruce Willis, Madeleine Stowe, Brad Pitt"
6 Exception: missing field
7 "Cold Mountain" "Anthony Minghella" "2003" "Adventure, Drama, History"
8 3 unique entries
9 search "Monkeys" in "title"
10 "Twelve Monkeys" "Terry Gilliam" "1995" "Mystery, Sci-Fi, Thriller" " Bruce Willis, Madeleine
  ↳ Stowe, Brad Pitt"
11 Exception: command is wrong
12 search "Spice" in "type"
13 sort "year"
14 "Twelve Monkeys" "Terry Gilliam" "1995" "Mystery, Sci-Fi, Thriller" " Bruce Willis, Madeleine
  ↳ Stowe, Brad Pitt"
15 "The Lord of the Rings: The Fellowship of the Ring" "Peter Jackson" "2001" "Adventure, Drama,
  ↳ Fantasy" "Elijah Wood, Ian McKellen, Orlando Bloom"
16 "Perfume: The Story of a Murderer" " Tom Tykwer" "2006" "Crime, Drama, Fantasy" "Ben Whishaw,
  ↳ Dustin Hoffman, Alan Rickman"
```

Remarks

- Be careful with the order of exceptions. If an entry has a missing field and it has the same first field with another entry, you should throw missing field exception.
- Assume no other errors will be present in the files.
- Try to generalize your program. (you can use templates).
- Efficiency is important. (try to use the existing (STL) containers and their methods for sorting etc...)

Turn in:

- Source code of a complete C++ program and a suitable makefile. You should use c++11 standard. Your code will be tested in a linux-gcc environment.
- A script will be used in order to check the correctness of your results. So, be careful not to violate the expected output format.
- Provide comments unless you are not interested in partial credit. (If I cannot easily understand your design, you may lose points.)
- You cannot get full credit if your implementation contradicts with the statements in this document.

Late Submission

- Not accepted.

Grading (Tentative)

- **Max Grade** : 100.
- Multiple tests(at least 5) will be performed.

All of the followings are possible deductions from **Max Grade**.

- Do **NOT** use hard-coded values. If you use you will loose 10pts.
- No submission: -100. (be consistent in doing this and your overall grade will converge to N/A) (To be specific: if you miss 3 assignments you'll get N/A)
- Compile errors: -100.
- Irrelevant code: -100.
- Major parts are missing: -100.
- Unnecessarily long code: -30.
- Inefficient implementation: -20.
- Using language elements and libraries which are not allowed: -100.
- Not caring about the structure and efficiency: -30. (avoid using hard-coded values, avoid hard-to-follow expressions, avoid code repetition, avoid unnecessary loops).
- Significant number of compiler warnings: -10.
- Not commented enough: -10. (Comments are in English. Turkish comments are not accepted).
- Source code encoding is not UTF-8 and characters are not properly displayed: -5. (You can use 'Visual Studio Code', 'Sublime Text', 'Atom' etc... Check the character encoding of your text editor and set it to UTF-8).
- Missing or wrong output values: **Fails the test**.
- Output format is wrong: -30.
- Infinite loop: **Fails the test**.
- Segmentation fault: **Fails the test**.
- Fails 5 or more random tests: -100.
- Fails the test: **deduction up to 20**.
- Prints anything extra: -30.
- Unwanted chars and spaces in output: -30.
- Submission includes files other than the expected: -10.
- Submission does not follow the file naming convention: -10.
- Sharing or inheriting code: -200.