GCS10004-07 Fall 2022

Handong Honor Code

- You are responsible for understanding and complying with Handong Honor Code.
- If you copy someone else's coding or homework assignments or use someone else's creative work without any indication, you will get an F grade. Anyone who shares the code with you will also get an F grade.

Jaeyoung Chun | School of Applied Artificial Intelligence | Handong Global University

Final Project

1 --

1.1 Create a PRIVATE Repository

Create a **PRIVATE** repository in your GitHub and name it as `my-bible-app`. You must create a **PRIVATE** repository. If you create a public repository and store your Bible app code there, you will get a **F** grade. I'm emphasizing this again - it must be a **PRIVATE** repository, otherwise you will get a **F** grade.

Please see `supplementary.pdf` for more information.

1.2 Add a Collaborator

Follow the instructions below to add `hisplan` to your `my-bible-app` GitHub repository as a collaborator so that `hisplan` can check and evaluate your final project:

1. Open a browser and go to your `my-bible-app` GitHub repository.
2. Click `Settings` on the top right corner.
3. Click `Collaborators` on the menu.
4. Click `Add people` on the bottom.
5. Type `hisplan` in the textbox and click `Select a collaborator above`.

Once done, email me with the following information, then I will let you know if this part of the final project is completed or not:

- Your Full Name
- Your Student ID
- Your GitHub URL (e.g. `https://github.com/****`)

Please see `supplementary.pdf` for more information.

2 --

2.1 Complete the Code

You are provided with `Bible.py`, a module that has partially implemented some core functionalities of a Bible search app. Your job is to finish the module by completing the missing code.

There are *six* places with the following placeholders where you need to write code:

```
### YOUR CODE STARTS HERE

### YOUR CODE ENDS HERE
```

The detailed instructions can be found in the code. So, please make sure you read the entire code before you proceed. This way, you will also get a big picture.

In `Bible.py`, you will see Q1, Q2, Q3, Q4, Q5, and Q6. Q1 through Q5 are mandatory. Q6 is an extra credit (+10 points)

`Bible.py` implements a class named `Bible` using Object Oriented Programming (OOP), so it has a constructor, methods and attributes. You should have a general understanding of OOP that we covered in the class in order to work with this code.

`Bible.py` has many `assert` statements that check whether you implement the code correctly or not. If any of these `assert` statements throw an exception, that means you did something wrong in the code, thus you should find the problems and fix them.

Note that `Bible.py` is a module, thus even if you did everything correctly, running `Bible.py` will not dislpay anything on the screen. Instead, you are provided with three testing code. By running any of these testing code, you will know whether you implemented `Bible.py` correctly or not.

- `test_search.py`
  - This will successfully run only when you complete Q1 through Q5 correctly.
- `test_max_chapter_verse.py`
  - This will successfully run only when you complete Q1 through Q5 correctly.
- `test_search_by_keyword.py`
  - This will successfully run only when you complete Q6 correctly (extra credit).

Please see `supplementary.pdf` for more information.

2.2 Create a Streamlit Web App

You are provided with `Bible_app.py` (a skeleton code). Your job is to implement the Bible search app using Streamlit and the `Bible.py` module that you have already completed in Section `2.1`.

In `supplementary.pdf`, you will find a screenshot of an example Bible search app that was demonstrated during class. Your app doesn't have to look like this. You can design the app however you want, but at minimum, the app should have UI elements (e.g. textbox, dropdown box) that allow users to search for a specific Bible verse by selecting a book name, chapter number, and verse number.

Make sure your name appear somewhere in the app.

2.3 Push Your Code to the Repository

Push your final code to the Github repository `my-bible-app`. The following is the list of files to be added to the GitHub repository:

```
Act.txt         Hag.txt         Lam.txt         Rom.txt
Amo.txt         Heb.txt         Lev.txt         Rut.txt
Ch1.txt         Hos.txt         Luk.txt         Sa1.txt
Ch2.txt         Isa.txt         Mal.txt         Sa2.txt
Co1.txt         Jam.txt         Mar.txt         Sol.txt
Co2.txt         Jde.txt         Mat.txt         Th1.txt
Col.txt         Jdg.txt         Mic.txt         Th2.txt
Dan.txt         Jer.txt         Nah.txt         Ti1.txt
Deu.txt         Jo1.txt         Neh.txt         Ti2.txt
Ecc.txt         Jo2.txt         Num.txt         Tit.txt
Eph.txt         Jo3.txt         Oba.txt         Zac.txt
Est.txt         Job.txt         Pe1.txt         Zep.txt
Exo.txt         Joe.txt         Pe2.txt         bible-3letter.txt
Eze.txt         Joh.txt         Phi.txt         bible-fullname.txt
Ezr.txt         Jon.txt         Plm.txt         max_chapter_info.txt
Gal.txt         Jos.txt         Pro.txt         max_verse_info.txt
Gen.txt         Kg1.txt         Psa.txt         Bible.py
Hab.txt         Kg2.txt         Rev.txt         Bible_app.py
```

2.4 Configure Streamlit Cloud

Configure Streamlit Cloud so that your Bible search app can be served to the public. Once done, include the web address of your Bible search app in the file named `site.txt`. For example,

```
http://***.streamlit.app
```

where `***` could be any name you chooose.

3 --

Another extra credit (+10 points): create a Tkinter-based Bible search app called `Bible_tkinter_app.py`.