**Bot.py**

1. Function: on\_ready

Description: run on bot start-up

1. Function: on\_guild\_join

Description: run when a user joins a guild with the bot present

Inputs:

-guild: the guild the user joined from

1. Function: on\_message

Description: run when a message is sent to a discord the bot occupies

Inputs:

-message: the message the user sent to a channel

1. Function: on\_message\_edit

Description: run when a user edits a message

Inputs:

- before: the old message

- after: the new message

1. Function: test  
   Description: Simple test command that shows commands are working.  
   Inputs:  
    - ctx: context of the command  
   Outputs:  
    - Sends test successful message back to channel that called test
2. Function: set\_instructor  
   Description: Command used to give Instructor role out by instructors  
   Inputs:  
    - ctx: context of the command  
    - member: user to give role  
   Outputs:  
    - Sends confirmation back to channel
3. Function: create\_event  
   Description: command to create event and send to event\_creation module  
   Ensures command author is Instructor  
   Inputs:  
    - ctx: context of the command  
   Outputs:  
    - Options to create event
4. Function: oh  
   Description: command related office hour and send to office\_hours module  
   Inputs:  
    - ctx: context of the command  
    - command: specific command to run  
    - \*args: arguments for command  
   Outputs:  
    - Office hour details and options
5. Function: ask  
   Description: command to ask question and sends to qna module  
   Inputs:  
    - ctx: context of the command  
    - question: question text  
   Outputs:  
    - User question in new post
6. Function: answer  
   Description: command to answer question and sends to qna module  
   Inputs:  
    - ctx: context of the command  
    - q\_num: question number to answer  
    - answer: answer text  
   Outputs:  
    - User answer in question post

# Function: begin\_tests  
# Description: Start the automated testing  
# Inputs:  
# - ctx: context of the command

# Function: end\_tests  
# Description: Finalize automated testing  
# Inputs:  
# - ctx: context of the command

# Function: test\_dummy  
# Description: Run the bot

Cal.py

Function: display\_events Description: Sends or updates the embed for the calendar Inputs: - ctx: context of function activation

Function: update\_calendar Description: Builds the calendar embed

Function: init Description: Initializes the calendar, creating channel and embed call Inputs: - b: bot

Event\_creation.py

Function: get\_times Description: helper function for acquiring the times an instructor wants event to be held during Inputs: - ctx: context of this discord message - event\_type: type of event which times are being asked for - command\_invoker: discord user who is creating event Outputs: the begin and end times for the event

Function: create\_event Description: creates an event by the specifications of the instructor creating the event Inputs: - ctx: context of this discord message - testing\_mode: flag indicating whether this event is being created during a system test Outputs: new event created in database

Function: init Description: initializes this module, giving it access to discord bot Inputs: - b: discord bot Outputs: None

Office Hours  
Class: Group Description: contains information about an office hour group

Class: OfficeHourQueue Description: contains information about an office hour queue

Method: enqueue Description: adds a student to the office hour queue Inputs: - student: student to add to the office hour queue Outputs: None

Method: display\_queue Description: displays the office hour queue in the office hour channel Outputs: office hour queue as a message in the office hour channel

Function: office\_hour\_command Description: handles a command given in an office hour channel Inputs: - ctx: context of this discord message - command: office hour command given - args: extra arguments given to command

Function: open\_oh Description: opens an office hour for students to get help from Inputs: - guild: discord guild this office hour is relevant for - ta: name of TA who is holding this office hour Outputs: creation of channels relevant to office hour

Function: close\_oh Description: closes an office hour session Inputs: - guild: discord guild this office hour is relevant for - ta: name of TA who is holding this office hour Outputs: deletion of channels relevant to office hour

Class: TaOfficeHour Description: contains information about when an office hour is held

Function: check\_office\_hour\_loop Description: runs intermittently to open or close office hours based on the current time

Function: add\_office\_hour Description: adds a new TA office hour to the guild Inputs: - guild: discord guild this office hour is relevant for - ta\_office\_hour: TA office hour information Outputs: adds a new TA office hour to the system

Function: init Description: initializes office hours module Inputs: - b: discord bot

Profanity.py

Function: check\_profanity Description: Uses better\_profanity to check profanity in a message Inputs: - msg: message from user

unction: censor\_profanity Description: censor the message per better\_profanity Inputs: - msg: message from user

qna.py

Class: QuestionsAnswers Description: object with question details Inputs: - q: question text - number: question number - message: id of the message associated with question - ans: answers to the question Outputs: None

Function: question Description: takes question from user and reposts anonymously and numbered Inputs: - ctx: context of the command - q: question text Outputs: - User question in new post

Function: answer Description: adds user answer to specific question and post anonymously Inputs: - ctx: context of the command - num: question number being answered - ans: answer text to question specified in num Outputs: - User answer added to question post