

Unit Tests for Auction Class

Assignment

1. Create a unit test class named **auction_test.py** to test that the Auction methods obey the rules of the auction and the behavior specified in the docstring comments for each method. This means you should try to verify the auction *behavior as well as auction methods*.
2. Use descriptive test method names such as **test_bid_when_auction_stopped**.
Not a descriptive name: **test_borderline**
3. Please **do not** write tests for obvious type errors such as **bid(None, "1.5")** where amount is not a number and name not a string. Type errors like that are the programmer's fault.
4. Use a **setUp** method to create an Auction object as a *test fixture* (an attribute) instead of creating an auction in each test. It is OK for some test methods to create their own auction object for a special test.

Evaluation

Your test code will be used to test several variations of the Auction class code. One variant is correct and the others all have a defect. Some may have multiple defects.

The auction variants are all tested on Github using a Github Action. The tests are run every time you "push" to Github. You can view the Action's **build output** to see which tests passed and failed, and the test output. But you cannot look at the source code for the variant Auction class.

Auction Specification

1. Each bid must have an amount > 0 and a non-blank bidder name (string). If not, a **ValueError** is raised.
2. The bid amount must be at least the current best bid *plus a minimum increment*. The minimum increment is specified as a parameter in the Auction constructor. with a default *minimum increment* is 1 (that is, a bid must be at least 1 more than the highest bid so far.) If bid is too low, an **AuctionError** is raised.
3. Bids allowed only when the auction is active. Auction has methods **start()** and **stop()** to enable and disable bidding, and **isActive()** to test if bidding is enabled. stop and start may be called many times during an auction to temporarily pause an auction.
4. The participant with the highest valid bid (best bid) wins.
5. The method **best_bid()** always returns the highest valid bid so far, and **winner()** returns the name of the bidder with the highest bid. These methods can be invoked at any time, even while the auction is active.
6. To mitigate accidental typing errors, the bidder name is always put into standard form. Surrounding white space is removed and words are converted to title case.
For example, if a bidder has name " donald DuCK ", the name will be converted to "Donald Duck".

Documentation

See the Docstring comments in code. To view them in a Python shell, use:

```
>>> from auction import Auction
```

```
>>> help(Auction)
```

```
>>> help(Auction.bid) - show docstring for the bid() method
```

What to Submit - *This will be explained in class.*

Push your auction test code and README.md file to Github.

In **README.md**, you must complete a table that describes each defect you find by testing, and try to identify the likely cause of the error. Since you can't look at the source code for the Auction variants, it isn't always possible to identify the cause of the defect.

Example Auction

You should not copy this example. Create your own test cases.

<code>auction = Auction("TDD in Python, 2E")</code>	Create auction for a great testing book!
<code>auction.start()</code>	Start accepting bids.
<code>auction.bid("Jim", 200)</code>	Jim bids 200 Baht.
<code>auction.bid("mai", 250)</code>	Mai bids 250 Baht.
<code>auction.bid("", 1000)</code> ValueError: Invalid bidder name	Anonymous bids are not allowed.
<code>auction.bid("Jim", 250.10)</code> AuctionError: Bid is too low	What a cheap-stake! Only 10 <i>satang</i> higher! But it is not allowed. The default bid increment is 1.
<code>auction.best_bid()</code> 250 <code>auction.winner()</code> 'Mai'	You can call these methods at any time to see the best bid so far and name of the top bidder. Names are normalized to Title Case.
<code>auction.bid("Jittat", 260)</code>	Ajarn Jittat wants the book, too!
<code>auction.stop()</code>	No bidding allowed now. Call start() to allow bidding.
<code>auction.bid("Jim", 265)</code> AuctionError: bidding not allowed now	Too late! Auction has stopped.
<code>auction.winner()</code> 'Jittat'	And Aj. Jittit gets " <i>TDD in Python</i> " at a bargain price!