Duration: **45 minutes**
Aids Allowed: ***None***

Student Number: |_|_|_|_|_|_|_|_|_|_|

Last (Family) Name(s): ______________________________

First (Given) Name(s): ______________________________

Do ***not*** *turn this page until you have received the signal to start.*
In the meantime, please read the instructions below *carefully*.

This term test consists of 3 questions on 12 pages (including this one), printed on both sides of the paper. *When you receive the signal to start, please make sure that your copy of the test is complete, fill in the identification section above, and write your name on the back of the last page.*

Answer each question directly on the test paper, in the space provided, and use one of the "blank" pages for rough work. If you need more space for one of your solutions, use a "blank" page and *indicate clearly the part of your work that should be marked.*

Marking Guide

# 1:	______/10
# 2:	______/15
# 3:	______/25
TOTAL:	______/50

Good Luck!

Use the space on this "blank" page for scratch work, or for any solution that did not fit elsewhere.
Clearly label each such solution with the appropriate question and part number.

Question 1. [10 marks]

Write the body of the function below so that it satisfies its docstring. Assume that module `stack.py` defines a class `Stack` that provides the usual methods: `is_empty()`, `push(item)`, `pop()`.

For full marks, your code must *not* depend on any details of the implementation of class `Stack`. In other words, the only thing you can do with a `Stack` object is to call some of its methods.

```
from stack import Stack

def size(stk):
    """(Stack) -> int
    Return the number of items on Stack stk, *without* modifying stk.
    (It's OK if the contents of stk are modified during the execution of this
    function, as long as everything is restored before the function returns.)
    """
    # Hint: You can use more than one stack.
```

Use the space on this "blank" page for scratch work, or for any solution that did not fit elsewhere.
Clearly label each such solution with the appropriate question and part number.

Question 2. [15 marks]

Write the body of the function below so that it satisfies its docstring Your code *must* be recursive.

```
def fib(n):
    """int -> int
    Return the nth fibonacci number.
    Where the fibonacci numbers are defined as: 1, 1, 2, 3, 5, 8, 13
    (each number is the sum of he two previous numbers)
    >>> fib(4)
    3
    """
```

Use the space on this "blank" page for scratch work, or for any solution that did not fit elsewhere.

Clearly label each such solution with the appropriate question and part number.

Question 3. [25 marks]

Write a series of classes, complete with documentation that satisfiy the following specification.

- A `building` has an `address` (an arbitrary string), and a number of rooms, provided at the time of construction.
- A `room` has a `name` (an arbitrary string) and a `square_footage` (float), provided at the time of construction.
- When printed, a `building` prints the sum of the square footages of all of its rooms
- A `house` is a type of `building` with at most 10 rooms, and prints `"Welcome to our house"`, plus the details of all of its rooms (name and square footage, separated by commas)
- If a `house` is created with too many rooms, a `BuildingCodeViolationError` should be raised
- A `business` may have any number of rooms, but no room may be named `Bedroom`, or have a square footage of less than 100, or else a `InvalidBusinessError` should be raised.
- It is possible to rename any room in any building (by specifying an old and a new name), but only a business can change the square footage of their rooms (by specifying the room name and the new square footage)
- Any invalid/improper input to any parameter (aside from those already mentioned) should raise a `BuildingCreationException`

Write a main code body (that should only execute when this file is run directly, not when it is imported), to perform the following:

- prompt the user for a type of building, address and number of rooms
  - *reminder: `input("prompt")` prompts the user for input and returns their response*
- prompt the user for room names and square footages until all rooms have been named
- if the user's inputs cause `BuildingCreationException` print `"oops..."`
- if the user's inputs cause any other type of error print `"you can't do that"`
- your main body code should not test the input, all decisions about what is/isn't valid input must be made by the classes you created in the first part of the question

Use the space on this "blank" page for scratch work, or for any solution that did not fit elsewhere.
Clearly label each such solution with the appropriate question and part number.

Use the space on this "blank" page for scratch work, or for any solution that did not fit elsewhere.
Clearly label each such solution with the appropriate question and part number.

Use the space on this "blank" page for scratch work, or for any solution that did not fit elsewhere.
Clearly label each such solution with the appropriate question and part number.

Use the space on this "blank" page for scratch work, or for any solution that did not fit elsewhere.
Clearly label each such solution with the appropriate question and part number.

On this page, please write nothing except your name.

Last (Family) Name(s): ____________________

First (Given) Name(s): ____________________

Total Marks = 50