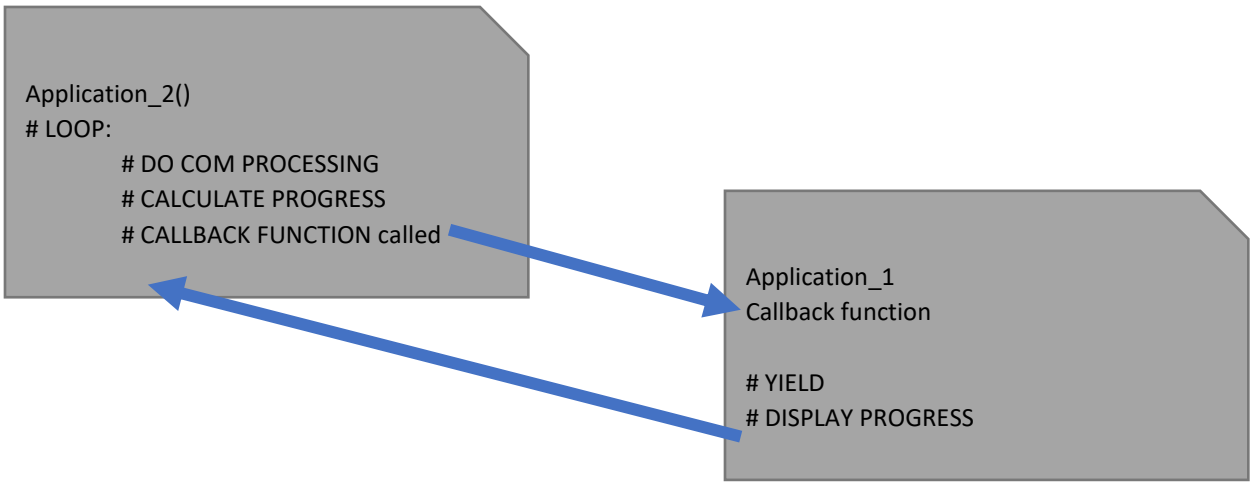
FF challenge Question #7



Using a generator as a callback function

```
def application_2(data, callback=None):
    if callback is not None:
        next(callback)

    for i, item in enumerate(data):
      # do com processing here
        process_com(item)
        if callback is not None:
            callback.send(float(i) / len(data))

    if callback is not None:  # Close the generator
        callback.close()

def callback(report_frequency):
    counter = 0
    while True:
        progress = yield
        counter += 1
        if counter % report_frequency:
            print("Progress: %s%%" % (progress * 100.0,))


# application_1
if __name__ == '__main__':
    # 'data' to process
    application_2 (data, callback(10))

#  By passing a callable as a parameter to application_2() function, a block of
# code can be executed, and if you yield from the callback function you can
# separate the COM processing from PROGRESS tracking. This way we are not
# exchanging any variables directly between application 1 and 2.
```