

```
"""
```

```
Bank waiting example
```

```
Scenario:
```

```
One or more counters with a fixed service time and customers who randomly arrive.
```

```
"""
```

```
import random
import simpy
```

```
RANDOM_SEED          = 42      # Needed to get the pseudo random
generator going
NEW_CUSTOMERS        = 10      # Total number of customers
INTERVAL_CUSTOMERS   = 9.0     # Generate new customers roughly every x
seconds
BANK_COUNTERS        = 1       # Number of counters
SERVICE_TIME         = 12.0    # Avg. time needed at the counter
```

```
""" Definition of the customer generator """
```

```
def customerGenerator(env, numberOfCustomers, interval,
bank_counter):
    """Source generates n customers each at a random time"""
    for i in range(numberOfCustomers):
        c = customer(env, 'Customer_%02d' % i, bank_counter,
SERVICE_TIME )
        env.process(c)
        t = random.normalvariate(interval, 1.0)
        yield env.timeout(t)
```

```
""" Definition of the customer """
```

```
def customer(env, name, bank_counter, time_at_counter):
    """Customer arrives, is served and leaves."""
    arrive = env.now
    print('Time: %7.4f, %s Arrived.' % (arrive, name))

    with bank_counter.request() as req:
        # Wait for the counter
        yield req
        wait = env.now - arrive
        # We got to the counter
        print('Time: %7.4f, %s Waited: %6.3f' % (env.now, name,
wait))
        t= time_at_counter
        t = random.normalvariate(time_at_counter, 2.0)
        yield env.timeout(t)
        print('Time: %7.4f, %s Finished' % (env.now, name))
```

```
# Setup and start the simulation
```

```
print('Bank waiting time simulation.')
random.seed(RANDOM_SEED)
simEnvironment = simpy.Environment()
```

```
# Start processes and run
bank_counter = simpy.Resource(simEnvironment, capacity=
BANK_COUNTERS)
generated_customers = customerGenerator(simEnvironment,
NEW_CUSTOMERS, INTERVAL_CUSTOMERS, bank_counter)
simEnvironment.process(generated_customers)
simEnvironment.run()
```