envvar1.py (Page 1 of 1)

```
 1: #!/usr/bin/env python
 2:
 3: #-----------------------------------------------------------------------
 4: # envvar1.py
 5: # Author: Bob Dondero
 6: #-----------------------------------------------------------------------
 7:
 8: import os
 9:
10: GREET = os.environ.get('GREETING', 'hi')
11:
12: def main():
13:     print(GREET)
14:
15: if __name__ == '__main__':
16:     main()
```

blank (Page 1 of 1)

1: This page is intentionally blank.

.env (Page 1 of 1)

```
1: GREETING=hello
```

envvar2.py (Page 1 of 1)

```python
 1: #!/usr/bin/env python
 2:
 3: #----------------------------------------------------------------------
 4: # envvar2.py
 5: # Author: Bob Dondero
 6: #----------------------------------------------------------------------
 7:
 8: import os
 9: import dotenv
10:
11: dotenv.load_dotenv()
12: GREET = os.environ.get('GREETING', 'hi')
13:
14: def main():
15:     print(GREET)
16:
17: if __name__ == '__main__':
18:     main()
```

daytimeiodelay/daytimeclient.py (Page 1 of 1)

```
 1: #!/usr/bin/env python
 2:
 3: #-----------------------------------------------------------------------
 4: # daytimeclient.py
 5: # Author: Bob Dondero
 6: #-----------------------------------------------------------------------
 7: # Try this:
 8: # daytimeclient.py time-a.nist.gov 13
 9: #-----------------------------------------------------------------------
10:
11: import sys
12: import socket
13:
14: #-----------------------------------------------------------------------
15:
16: def main():
17:
18:     if len(sys.argv) != 3:
19:         print('Usage: python %s host port' % sys.argv[0])
20:         sys.exit(1)
21:
22:     try:
23:         host = sys.argv[1]
24:         port = int(sys.argv[2])
25:
26:         with socket.socket() as sock:
27:             sock.connect((host, port))
28:             flo = sock.makefile(mode='r', encoding='ascii')
29:             for line in flo:
30:                 print(line, end='')
31:
32:     except Exception as ex:
33:         print(ex, file=sys.stderr)
34:         sys.exit(1)
35:
36: #-----------------------------------------------------------------------
37:
38: if __name__ == '__main__':
39:     main()
```

blank (Page 1 of 1)

```
 1: This page is intentionally blank.
```

daytimeiodelay/daytimeserver.py (Page 1 of 2)

```
 1: #!/usr/bin/env python
 2:
 3: #-----------------------------------------------------------------------
 4: # daytimeserver.py
 5: # Author: Bob Dondero
 6: #-----------------------------------------------------------------------
 7:
 8: import os
 9: import sys
10: import socket
11: import time
12:
13: #-----------------------------------------------------------------------
14:
15: IODELAY = int(os.environ.get('IODELAY', '0'))
16:
17: #-----------------------------------------------------------------------
18:
19: def handle_client(sock):
20:
21:     # Simulate an I/O-bound server.
22:     time.sleep(IODELAY)
23:
24:     datetime = time.asctime(time.localtime())
25:     flo = sock.makefile(mode='w', encoding='ascii')
26:     flo.write(datetime + '\n')
27:     flo.flush()
28:
29: #-----------------------------------------------------------------------
30:
31: def main():
32:
33:     if len(sys.argv) != 2:
34:         print('Usage: python %s port' % sys.argv[0])
35:         sys.exit(1)
36:
37:     try:
38:         port = int(sys.argv[1])
39:
40:         server_sock = socket.socket()
41:         print('Opened server socket')
42:         if os.name != 'nt':
43:             server_sock.setsockopt(
44:                 socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
45:         server_sock.bind(('', port))
46:         print('Bound server socket to port')
47:         server_sock.listen()
48:         print('Listening')
49:
50:         while True:
51:             try:
52:                 sock, _ = server_sock.accept()
53:                 with sock:
54:                     print('Accepted connection')
55:                     print('Opened socket')
56:                     handle_client(sock)
57:                 print('Closed socket')
58:             except Exception as ex:
59:                 print(ex, file=sys.stderr)
60:
61:     except Exception as ex:
62:         print(ex, file=sys.stderr)
63:         sys.exit(1)
64:
65: #-----------------------------------------------------------------------
```

daytimeiodelay/daytimeserver.py (Page 2 of 2)

```
66:
67: if __name__ == '__main__':
68:     main()
```

daytimeiodelayp/daytimeserver.py (Page 1 of 2)

```python
 1: #!/usr/bin/env python
 2:
 3: #-----------------------------------------------------------------------------
 4: # daytimeserver.py
 5: # Author: Bob Dondero
 6: #-----------------------------------------------------------------------------
 7:
 8: import os
 9: import sys
10: import socket
11: import time
12: import multiprocessing
13:
14: #-----------------------------------------------------------------------------
15:
16: IODELAY = int(os.environ.get('IODELAY', '0'))
17:
18: #-----------------------------------------------------------------------------
19:
20: def handle_client(sock):
21:
22:     print('Forked child process')
23:
24:     try:
25:
26:         # Simulate an I/O-bound server.
27:         time.sleep(IODELAY)
28:
29:         with sock:
30:             datetime = time.asctime(time.localtime())
31:             flo = sock.makefile(mode='w', encoding='ascii')
32:             flo.write(datetime + '\n')
33:             flo.flush()
34:
35:         print('Closed socket in child process')
36:         print('Exiting child process')
37:
38:     except Exception as ex:
39:         print(ex, file=sys.stderr)
40:         sys.exit(1)
41:
42: #-----------------------------------------------------------------------------
43:
44: def main():
45:
46:     if len(sys.argv) != 2:
47:         print('Usage: python %s port' % sys.argv[0])
48:         sys.exit(1)
49:
50:     try:
51:         port = int(sys.argv[1])
52:
53:         server_sock = socket.socket()
54:         print('Opened server socket')
55:         if os.name != 'nt':
56:             server_sock.setsockopt(
57:                 socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
58:         server_sock.bind(('', port))
59:         print('Bound server socket to port')
60:         server_sock.listen()
61:         print('Listening')
62:
63:         while True:
64:             sock, _ = server_sock.accept()
65:             with sock:
```

daytimeiodelayp/daytimeserver.py (Page 2 of 2)

```python
66:                 print('Accepted connection')
67:                 print('Opened socket')
68:                 process = multiprocessing.Process(
69:                     target=handle_client,
70:                     args=[sock])
71:                 process.start()
72:             print('Closed socket in parent process')
73:
74:     except Exception as ex:
75:         print(ex, file=sys.stderr)
76:         sys.exit(1)
77:
78: #-----------------------------------------------------------------------------
79:
80: if __name__ == '__main__':
81:     main()
```

daytimeiodelayt/daytimeserver.py (Page 1 of 2)

```
 1: #!/usr/bin/env python
 2:
 3: #-----------------------------------------------------------------------
 4: # daytimeserver.py
 5: # Author: Bob Dondero
 6: #-----------------------------------------------------------------------
 7:
 8: import os
 9: import sys
10: import socket
11: import time
12: import threading
13:
14: #-----------------------------------------------------------------------
15:
16: IODELAY = int(os.environ.get('IODELAY', '0'))
17:
18: #-----------------------------------------------------------------------
19:
20: class ClientHandlerThread (threading.Thread):
21:
22:     def __init__(self, sock):
23:         threading.Thread.__init__(self)
24:         self._sock = sock
25:
26:     def run(self):
27:         print('Spawned child thread')
28:
29:         # Simulate an I/O-bound server.
30:         time.sleep(IODELAY)
31:
32:         with self._sock:
33:             datetime = time.asctime(time.localtime())
34:             flo = self._sock.makefile(mode='w', encoding='ascii')
35:             flo.write(datetime + '\n')
36:             flo.flush()
37:
38:         print('Closed socket in child thread')
39:         print('Exiting child thread')
40:
41: #-----------------------------------------------------------------------
42:
43: def main():
44:
45:     if len(sys.argv) != 2:
46:         print('Usage: python %s port' % sys.argv[0])
47:         sys.exit(1)
48:
49:     try:
50:         port = int(sys.argv[1])
51:
52:         server_sock = socket.socket()
53:         print('Opened server socket')
54:         if os.name != 'nt':
55:             server_sock.setsockopt(
56:                 socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
57:         server_sock.bind(('', port))
58:         print('Bound server socket to port')
59:         server_sock.listen()
60:         print('Listening')
61:
62:         while True:
63:             try:
64:                 sock, _ = server_sock.accept()
65:                 print('Accepted connection')
```

daytimeiodelayt/daytimeserver.py (Page 2 of 2)

```
66:                 print('Opened socket')
67:                 client_handler_thread = ClientHandlerThread(sock)
68:                 client_handler_thread.start()
69:             except Exception as ex:
70:                 print(ex, file=sys.stderr)
71:
72:     except Exception as ex:
73:         print(ex, file=sys.stderr)
74:         sys.exit(1)
75:
76: #-----------------------------------------------------------------------
77:
78: if __name__ == '__main__':
79:     main()
```

daytimecdelay/daytimeserver.py (Page 1 of 2)

```
 1: #!/usr/bin/env python
 2:
 3: #-----------------------------------------------------------------------
 4: # daytimeserver.py
 5: # Author: Bob Dondero
 6: #-----------------------------------------------------------------------
 7:
 8: import os
 9: import sys
10: import socket
11: import time
12:
13: #-----------------------------------------------------------------------
14:
15: CDELAY = int(os.environ.get('CDELAY', '0'))
16:
17: #-----------------------------------------------------------------------
18:
19: def consume_cpu_time(delay):
20:
21:     initial_thread_time = time.thread_time()
22:     while (time.thread_time() - initial_thread_time) < delay:
23:         pass
24:
25: #-----------------------------------------------------------------------
26:
27: def handle_client(sock):
28:
29:     # Simulate a compute-bound server.
30:     consume_cpu_time(CDELAY)
31:
32:     datetime = time.asctime(time.localtime())
33:     flo = sock.makefile(mode='w', encoding='ascii')
34:     flo.write(datetime + '\n')
35:     flo.flush()
36:
37: #-----------------------------------------------------------------------
38:
39: def main():
40:
41:     if len(sys.argv) != 2:
42:         print('Usage: python %s port' % sys.argv[0])
43:         sys.exit(1)
44:
45:     try:
46:         port = int(sys.argv[1])
47:
48:         server_sock = socket.socket()
49:         print('Opened server socket')
50:         if os.name != 'nt':
51:             server_sock.setsockopt(
52:                 socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
53:         server_sock.bind(('', port))
54:         print('Bound server socket to port')
55:         server_sock.listen()
56:         print('Listening')
57:
58:         while True:
59:             try:
60:                 sock, _ = server_sock.accept()
61:                 with sock:
62:                     print('Accepted connection')
63:                     print('Opened socket')
64:                     handle_client(sock)
65:                 print('Closed socket')
```

daytimecdelay/daytimeserver.py (Page 2 of 2)

```
66:             except Exception as ex:
67:                 print(ex, file=sys.stderr)
68:
69:     except Exception as ex:
70:         print(ex, file=sys.stderr)
71:         sys.exit(1)
72:
73: #-----------------------------------------------------------------------
74:
75: if __name__ == '__main__':
76:     main()
```

daytimecdelayp/daytimeserver.py (Page 1 of 2)

```
 1: #!/usr/bin/env python
 2:
 3: #-----------------------------------------------------------------------
 4: # daytimeserver.py
 5: # Author: Bob Dondero
 6: #-----------------------------------------------------------------------
 7:
 8: import os
 9: import sys
10: import socket
11: import time
12: import multiprocessing
13:
14: #-----------------------------------------------------------------------
15:
16: CDELAY = int(os.environ.get('CDELAY', '0'))
17:
18: #-----------------------------------------------------------------------
19:
20: def consume_cpu_time(delay):
21:
22:     initial_thread_time = time.thread_time()
23:     while (time.thread_time() - initial_thread_time) < delay:
24:         pass
25:
26: #-----------------------------------------------------------------------
27:
28: def handle_client(sock):
29:
30:     print('Forked child process')
31:
32:     try:
33:
34:         # Simulate a compute-bound server.
35:         consume_cpu_time(CDELAY)
36:
37:         With sock:
38:             datetime = time.asctime(time.localtime())
39:             flo = sock.makefile(mode='w', encoding='ascii')
40:             flo.write(datetime + '\n')
41:             flo.flush()
42:
43:         print('Closed socket in child process')
44:         print('Exiting child process')
45:
46:     except Exception as ex:
47:         print(ex, file=sys.stderr)
48:         sys.exit(1)
49:
50: #-----------------------------------------------------------------------
51:
52: def main():
53:
54:     if len(sys.argv) != 2:
55:         print('Usage: python %s port' % sys.argv[0])
56:         sys.exit(1)
57:
58:     try:
59:         port = int(sys.argv[1])
60:
61:         server_sock = socket.socket()
62:         print('Opened server socket')
63:         if os.name != 'nt':
64:             server_sock.setsockopt(
65:                 socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
```

daytimecdelayp/daytimeserver.py (Page 2 of 2)

```
66:         server_sock.bind(('', port))
67:         print('Bound server socket to port')
68:         server_sock.listen()
69:         print('Listening')
70:
71:         while True:
72:             sock, _ = server_sock.accept()
73:             with sock:
74:                 print('Accepted connection')
75:                 print('Opened socket')
76:                 process = multiprocessing.Process(
77:                     target=handle_client,
78:                     args=[sock])
79:                 process.start()
80:             print('Closed socket in parent process')
81:
82:     except Exception as ex:
83:         print(ex, file=sys.stderr)
84:         sys.exit(1)
85:
86: #-----------------------------------------------------------------------
87:
88: if __name__ == '__main__':
89:     main()
```

daytimecdelayt/daytimeserver.py (Page 1 of 2)

```
 1: #!/usr/bin/env python
 2:
 3: #-----------------------------------------------------------------------
 4: # daytimeserver.py
 5: # Author: Bob Dondero
 6: #-----------------------------------------------------------------------
 7:
 8: import os
 9: import sys
10: import socket
11: import time
12: import threading
13:
14: #-----------------------------------------------------------------------
15:
16: CDELAY = int(os.environ.get('CDELAY', '0'))
17:
18: #-----------------------------------------------------------------------
19:
20: def consume_cpu_time(delay):
21:
22:     initial_thread_time = time.thread_time()
23:     while (time.thread_time() - initial_thread_time) < delay:
24:         pass
25:
26: #-----------------------------------------------------------------------
27:
28: class ClientHandlerThread (threading.Thread):
29:
30:     def __init__(self, sock):
31:         threading.Thread.__init__(self)
32:         self._sock = sock
33:
34:     def run(self):
35:         print('Spawned child thread')
36:
37:         # Simulate a compute-bound server.
38:         consume_cpu_time(CDELAY)
39:
40:         with self._sock:
41:             datetime = time.asctime(time.localtime())
42:             flo = self._sock.makefile(mode='w', encoding='ascii')
43:             flo.write(datetime + '\n')
44:             flo.flush()
45:
46:         print('Closed socket in child thread')
47:         print('Exiting child thread')
48:
49: #-----------------------------------------------------------------------
50:
51: def main():
52:
53:     if len(sys.argv) != 2:
54:         print('Usage: python %s port' % sys.argv[0])
55:         sys.exit(1)
56:
57:     try:
58:         port = int(sys.argv[1])
59:
60:         server_sock = socket.socket()
61:         print('Opened server socket')
62:         if os.name != 'nt':
63:             server_sock.setsockopt(
64:                 socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
65:         server_sock.bind(('', port))
```

daytimecdelayt/daytimeserver.py (Page 2 of 2)

```
66:         print('Bound server socket to port')
67:         server_sock.listen()
68:         print('Listening')
69:
70:         while True:
71:             try:
72:                 sock, _ = server_sock.accept()
73:                 print('Accepted connection')
74:                 print('Opened socket')
75:                 client_handler_thread = ClientHandlerThread(sock)
76:                 client_handler_thread.start()
77:             except Exception as ex:
78:                 print(ex, file=sys.stderr)
79:
80:     except Exception as ex:
81:         print(ex, file=sys.stderr)
82:         sys.exit(1)
83:
84: #-----------------------------------------------------------------------
85:
86: if __name__ == '__main__':
87:     main()
```

deadlock.py (Page 1 of 2)

```
 1: #!/usr/bin/env python
 2:
 3: #-----------------------------------------------------------------------
 4: # deadlock.py
 5: # Author: Bob Dondero
 6: #-----------------------------------------------------------------------
 7:
 8: import threading
 9:
10: #-----------------------------------------------------------------------
11:
12: class BankAcct:
13:
14:     def __init__(self, title):
15:         self._title = title
16:         self._balance = 0
17:         self._lock = threading.RLock()
18:
19:     def transfer_to(self, other, amount):
20:         self._lock.acquire()
21:         other._lock.acquire()
22:         self._balance -= amount
23:         other._balance += amount
24:         print(self._title + ': ' + str(self._balance))
25:         print(other._title + ': ' + str(other._balance))
26:         other._lock.release()
27:         self._lock.release()
28:
29: #-----------------------------------------------------------------------
30:
31: class AliceToBobThread (threading.Thread):
32:
33:     def __init__(self, alice_acct, bob_acct):
34:         threading.Thread.__init__(self)
35:         self._alice_acct = alice_acct
36:         self._bob_acct = bob_acct
37:
38:     def run(self):
39:         for _ in range(1000):
40:             self._alice_acct.transfer_to(self._bob_acct, 1)
41:
42: #-----------------------------------------------------------------------
43:
44: class BobToAliceThread (threading.Thread):
45:
46:     def __init__(self, bob_acct, alice_acct):
47:         threading.Thread.__init__(self)
48:         self._bob_acct = bob_acct
49:         self._alice_acct = alice_acct
50:
51:     def run(self):
52:         for _ in range(1000):
53:             self._bob_acct.transfer_to(self._alice_acct, 1)
54:
55: #-----------------------------------------------------------------------
56:
57: def main():
58:
59:     alice_acct = BankAcct('Alice')
60:     bob_acct = BankAcct('Bob')
61:
62:     alice_to_bob_thread = AliceToBobThread(alice_acct, bob_acct)
63:     bob_to_alice_thread = BobToAliceThread(bob_acct, alice_acct)
64:
65:     alice_to_bob_thread.start()
```

deadlock.py (Page 2 of 2)

```
66:     bob_to_alice_thread.start()
67:
68:     alice_to_bob_thread.join()
69:     bob_to_alice_thread.join()
70:
71:     print('Finished')
72:
73: #-----------------------------------------------------------------------
74:
75: if __name__ == '__main__':
76:     main()
```

deadlockw.py (Page 1 of 2)

```
 1: #!/usr/bin/env python
 2:
 3: #-----------------------------------------------------------------------
 4: # deadlockw.py
 5: # Author: Bob Dondero
 6: #-----------------------------------------------------------------------
 7:
 8: import threading
 9:
10: #-----------------------------------------------------------------------
11:
12: class BankAcct:
13:
14:     def __init__(self, title):
15:         self._title = title
16:         self._balance = 0
17:         self._lock = threading.RLock()
18:
19:     def transfer_to(self, other, amount):
20:         with self._lock:
21:             with other._lock:
22:                 self._balance -= amount
23:                 other._balance += amount
24:                 print(self._title + ': ' + str(self._balance))
25:                 print(other._title + ': ' + str(other._balance))
26:
27: #-----------------------------------------------------------------------
28:
29: class AliceToBobThread (threading.Thread):
30:
31:     def __init__(self, alice_acct, bob_acct):
32:         threading.Thread.__init__(self)
33:         self._alice_acct = alice_acct
34:         self._bob_acct = bob_acct
35:
36:     def run(self):
37:         for _ in range(1000):
38:             self._alice_acct.transfer_to(self._bob_acct, 1)
39:
40: #-----------------------------------------------------------------------
41:
42: class BobToAliceThread (threading.Thread):
43:
44:     def __init__(self, bob_acct, alice_acct):
45:         threading.Thread.__init__(self)
46:         self._bob_acct = bob_acct
47:         self._alice_acct = alice_acct
48:
49:     def run(self):
50:         for _ in range(1000):
51:             self._bob_acct.transfer_to(self._alice_acct, 1)
52:
53: #-----------------------------------------------------------------------
54:
55: def main():
56:
57:     alice_acct = BankAcct('Alice')
58:     bob_acct = BankAcct('Bob')
59:
60:     alice_to_bob_thread = AliceToBobThread(alice_acct, bob_acct)
61:     bob_to_alice_thread = BobToAliceThread(bob_acct, alice_acct)
62:
63:     alice_to_bob_thread.start()
64:     bob_to_alice_thread.start()
65:
```

deadlockw.py (Page 2 of 2)

```
66:     alice_to_bob_thread.join()
67:     bob_to_alice_thread.join()
68:
69:     print('Finished')
70:
71: #-----------------------------------------------------------------------
72:
73: if __name__ == '__main__':
74:     main()
```

nodeadlock.py (Page 1 of 2)

```
 1: #!/usr/bin/env python
 2:
 3: #---------------------------------------------------------------------
 4: # nodeadlock.py
 5: # Author: Bob Dondero
 6: #---------------------------------------------------------------------
 7:
 8: import threading
 9:
10: #---------------------------------------------------------------------
11:
12: class BankAcct:
13:
14:     def __init__(self, title, sequence_num):
15:         self._title = title
16:         self._balance = 0
17:         self._sequence_num = sequence_num
18:         self._lock = threading.RLock()
19:
20:     def transfer_to(self, other, amount):
21:         if self._sequence_num < other._sequence_num:
22:             self._lock.acquire()
23:             other._lock.acquire()
24:             self._balance -= amount
25:             other._balance += amount
26:             print(self._title + ': ' + str(self._balance))
27:             print(other._title + ': ' + str(other._balance))
28:             other._lock.release()
29:             self._lock.release()
30:         else:
31:             other._lock.acquire()
32:             self._lock.acquire()
33:             self._balance -= amount
34:             other._balance += amount
35:             print(self._title + ': ' + str(self._balance))
36:             print(other._title + ': ' + str(other._balance))
37:             self._lock.release()
38:             other._lock.release()
39:
40: #---------------------------------------------------------------------
41:
42: class AliceToBobThread (threading.Thread):
43:
44:     def __init__(self, alice_acct, bob_acct):
45:         threading.Thread.__init__(self)
46:         self._alice_acct = alice_acct
47:         self._bob_acct = bob_acct
48:
49:     def run(self):
50:         for _ in range(1000):
51:             self._alice_acct.transfer_to(self._bob_acct, 1)
52:
53: #---------------------------------------------------------------------
54:
55: class BobToAliceThread (threading.Thread):
56:
57:     def __init__(self, bob_acct, alice_acct):
58:         threading.Thread.__init__(self)
59:         self._bob_acct = bob_acct
60:         self._alice_acct = alice_acct
61:
62:     def run(self):
63:         for _ in range(1000):
64:             self._bob_acct.transfer_to(self._alice_acct, 1)
65:
```

nodeadlock.py (Page 2 of 2)

```
66: #---------------------------------------------------------------------
67:
68: def main():
69:
70:     alice_acct = BankAcct('Alice', 1)
71:     bob_acct = BankAcct('Bob', 2)
72:
73:     alice_to_bob_thread = AliceToBobThread(alice_acct, bob_acct)
74:     bob_to_alice_thread = BobToAliceThread(bob_acct, alice_acct)
75:
76:     alice_to_bob_thread.start()
77:     bob_to_alice_thread.start()
78:
79:     alice_to_bob_thread.join()
80:     bob_to_alice_thread.join()
81:
82:     print('Finished')
83:
84: #---------------------------------------------------------------------
85:
86: if __name__ == '__main__':
87:     main()
```

nodeadlockw.py (Page 1 of 2)

```
 1: #!/usr/bin/env python
 2:
 3: #-----------------------------------------------------------------------
 4: # nodeadlockw.py
 5: # Author: Bob Dondero
 6: #-----------------------------------------------------------------------
 7:
 8: import threading
 9:
10: #-----------------------------------------------------------------------
11:
12: class BankAcct:
13:
14:     def __init__(self, title, sequence_num):
15:         self._title = title
16:         self._balance = 0
17:         self._sequence_num = sequence_num
18:         self._lock = threading.RLock()
19:
20:     def transfer_to(self, other, amount):
21:         if self._sequence_num < other._sequence_num:
22:             with self._lock:
23:                 with other._lock:
24:                     self._balance -= amount
25:                     other._balance += amount
26:                     print(self._title + ': ' + str(self._balance))
27:                     print(other._title + ': ' + str(other._balance))
28:         else:
29:             with other._lock:
30:                 with self._lock:
31:                     self._balance -= amount
32:                     other._balance += amount
33:                     print(self._title + ': ' + str(self._balance))
34:                     print(other._title + ': ' + str(other._balance))
35:
36: #-----------------------------------------------------------------------
37:
38: class AliceToBobThread (threading.Thread):
39:
40:     def __init__(self, alice_acct, bob_acct):
41:         threading.Thread.__init__(self)
42:         self._alice_acct = alice_acct
43:         self._bob_acct = bob_acct
44:
45:     def run(self):
46:         for _ in range(1000):
47:             self._alice_acct.transfer_to(self._bob_acct, 1)
48:
49: #-----------------------------------------------------------------------
50:
51: class BobToAliceThread (threading.Thread):
52:
53:     def __init__(self, bob_acct, alice_acct):
54:         threading.Thread.__init__(self)
55:         self._bob_acct = bob_acct
56:         self._alice_acct = alice_acct
57:
58:     def run(self):
59:         for _ in range(1000):
60:             self._bob_acct.transfer_to(self._alice_acct, 1)
61:
62: #-----------------------------------------------------------------------
63:
64: def main():
65:
```

nodeadlockw.py (Page 2 of 2)

```
66:     alice_acct = BankAcct('Alice', 1)
67:     bob_acct = BankAcct('Bob', 2)
68:
69:     alice_to_bob_thread = AliceToBobThread(alice_acct, bob_acct)
70:     bob_to_alice_thread = BobToAliceThread(bob_acct, alice_acct)
71:
72:     alice_to_bob_thread.start()
73:     bob_to_alice_thread.start()
74:
75:     alice_to_bob_thread.join()
76:     bob_to_alice_thread.join()
77:
78:     print('Finished')
79:
80: #-----------------------------------------------------------------------
81:
82: if __name__ == '__main__':
83:     main()
```