

commchars/client.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # client.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sys
9: import socket
10:
11: #-----
12:
13: def write_courses(courses):
14:     for course in courses:
15:         print(course['name'])
16:         book = course['book']
17:         print(book['title'])
18:         print(book['price'])
19:     print()
20:
21: #-----
22:
23: def main():
24:     if len(sys.argv) != 3:
25:         print('Usage: python %s host port' % sys.argv[0])
26:         sys.exit(1)
27:     try:
28:         host = sys.argv[1]
29:         port = int(sys.argv[2])
30:
31:         with socket.socket() as sock:
32:             sock.connect((host, port))
33:             with sock.makefile(mode='r', encoding='utf-8') as flo:
34:                 courses = []
35:
36:                 while True:
37:                     course_name = flo.readline()
38:                     if course_name == '':
39:                         break
40:                     course_name = course_name.rstrip()
41:
42:                     book_title = flo.readline()
43:                     if book_title == '':
44:                         raise Exception('Ran out of input')
45:                     book_title = book_title.rstrip()
46:
47:                     book_price = flo.readline()
48:                     if book_price == '':
49:                         raise Exception('Ran out of input')
50:                     book_price = float(book_price.rstrip())
51:
52:                     book = {'title': book_title, 'price': book_price}
53:                     course = {'name': course_name, 'book': book}
54:                     courses.append(course)
55:
56:                 write_courses(courses)
57:
58:     except Exception as ex:
59:         print(ex, file=sys.stderr)
60:         sys.exit(1)
61:
62: #-----
63: if __name__ == '__main__':
64:     main()

```

commchars/server.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # server.py
5: # Author: Bob Dondero
6: #-----
7:
8: import os
9: import sys
10: import socket
11:
12: #-----
13:
14: def create_courses():
15:     book0 = {'title': 'C Programming', 'price': 88.55}
16:     book1 = {'title': 'The Practice of Programming', 'price': 35.14}
17:     course0 = {'name': 'COS 217', 'book': book0}
18:     course1 = {'name': 'COS 333', 'book': book1}
19:     courses = [course0, course1]
20:     return courses
21:
22: #-----
23:
24: def handle_client(sock):
25:     courses = create_courses()
26:     with sock.makefile(mode='w', encoding='utf-8') as flo:
27:         for course in courses:
28:             flo.write(course['name'] + '\n')
29:             book = course['book']
30:             flo.write(book['title'] + '\n')
31:             flo.write(str(book['price']) + '\n')
32:             #flo.flush()
33:         print('Wrote to client')
34:
35: #-----
36:
37: def main():
38:     if len(sys.argv) != 2:
39:         print('Usage: python %s port' % sys.argv[0])
40:         sys.exit(1)
41:     try:
42:         port = int(sys.argv[1])
43:         server_sock = socket.socket()
44:         print('Opened server socket')
45:         if os.name != 'nt':
46:             server_sock.setsockopt(
47:                 socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
48:         server_sock.bind(('', port))
49:         server_sock.listen()
50:         while True:
51:             try:
52:                 sock, _ = server_sock.accept()
53:                 with sock:
54:                     print('Accepted connection')
55:                     handle_client(sock)
56:             except Exception as ex:
57:                 print(ex, file=sys.stderr)
58:         except Exception as ex:
59:             print(ex, file=sys.stderr)
60:             sys.exit(1)
61:
62: #-----
63: if __name__ == '__main__':
64:     main()

```

commjson/client.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # client.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sys
9: import socket
10: import json
11:
12: #-----
13:
14: def write_courses(courses):
15:
16:     for course in courses:
17:         print(course['name'])
18:         book = course['book']
19:         print(book['title'])
20:         print(book['price'])
21:         print()
22:
23: #-----
24:
25: def main():
26:
27:     if len(sys.argv) != 3:
28:         print('Usage: python %s host port' % sys.argv[0])
29:         sys.exit(1)
30:
31:     try:
32:         host = sys.argv[1]
33:         port = int(sys.argv[2])
34:
35:         with socket.socket() as sock:
36:             sock.connect((host, port))
37:             with sock.makefile(mode='r', encoding='utf-8') as flo:
38:                 json_str = flo.readline()
39:
40:             json_str = json_str.rstrip()
41:             courses = json.loads(json_str)
42:
43:             write_courses(courses)
44:
45:     except Exception as ex:
46:         print(ex, file=sys.stderr)
47:         sys.exit(1)
48:
49: #-----
50:
51: if __name__ == '__main__':
52:     main()

```

commjson/server.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # server.py
5: # Author: Bob Dondero
6: #-----
7:
8: import os
9: import sys
10: import socket
11: import json
12:
13: #-----
14:
15: def create_courses():
16:     book0 = {'title': 'C Programming', 'price': 88.55}
17:     book1 = {'title': 'The Practice of Programming', 'price': 35.14}
18:     course0 = {'name': 'COS 217', 'book': book0}
19:     course1 = {'name': 'COS 333', 'book': book1}
20:     courses = [course0, course1]
21:     return courses
22:
23: #-----
24:
25: def handle_client(sock):
26:     courses = create_courses()
27:     json_str = json.dumps(courses)
28:     with sock.makefile(mode='w', encoding='utf-8') as flo:
29:         flo.write(json_str + '\n')
30:         #flo.flush()
31:     print('Wrote to client')
32:
33: #-----
34:
35: def main():
36:     if len(sys.argv) != 2:
37:         print('Usage: python %s port' % sys.argv[0])
38:         sys.exit(1)
39:
40:     try:
41:         port = int(sys.argv[1])
42:         server_sock = socket.socket()
43:         print('Opened server socket')
44:         if os.name != 'nt':
45:             server_sock.setsockopt(
46:                 socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
47:         server_sock.bind(('', port))
48:         server_sock.listen()
49:         while True:
50:             try:
51:                 sock, _ = server_sock.accept()
52:                 with sock:
53:                     print('Accepted connection')
54:                     handle_client(sock)
55:             except Exception as ex:
56:                 print(ex, file=sys.stderr)
57:         except Exception as ex:
58:             print(ex, file=sys.stderr)
59:             sys.exit(1)
60:
61: #-----
62:
63: if __name__ == '__main__':
64:     main()

```

commjsonbad/client.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # client.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sys
9: import socket
10: import json
11:
12: #-----
13:
14: def write_courses(courses):
15:
16:     for course in courses:
17:         print(course['name'])
18:         book = course['book']
19:         print(book['title'])
20:         print(book['price'])
21:         print()
22:
23: #-----
24:
25: def main():
26:
27:     if len(sys.argv) != 3:
28:         print('Usage: python %s host port' % sys.argv[0])
29:         sys.exit(1)
30:
31:     try:
32:         host = sys.argv[1]
33:         port = int(sys.argv[2])
34:
35:         with socket.socket() as sock:
36:             sock.connect((host, port))
37:             with sock.makefile(mode='r', encoding='utf-8') as flo:
38:                 json_str = flo.readline()
39:
40:                 json_str = json_str.rstrip()
41:                 courses = json.loads(json_str)
42:
43:                 write_courses(courses)
44:
45:     except Exception as ex:
46:         print(ex, file=sys.stderr)
47:         sys.exit(1)
48:
49: #-----
50:
51: if __name__ == '__main__':
52:     main()

```

commjsonbad/server.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # server.py
5: # Author: Bob Dondero
6: #-----
7:
8: import os
9: import sys
10: import socket
11: import json
12:
13: #-----
14:
15: def create_courses():
16:     # book0 = {'title': 'C Programming', 'price': 88.55}
17:     book1 = {'title': 'The Practice of Programming', 'price': 35.14}
18:     course0 = {'name': 'COS 217', 'book': book1} # Shared book.
19:     course1 = {'name': 'COS 333', 'book': book1} # Shared book.
20:     courses = [course0, course1]
21:     return courses
22:
23: #-----
24:
25: def handle_client(sock):
26:     courses = create_courses()
27:     json_str = json.dumps(courses)
28:     json_line = json_str + '\n'
29:     with sock.makefile(mode='w', encoding='utf-8') as flo:
30:         flo.write(json_line)
31:         #flo.flush()
32:     print('Wrote to client')
33:
34: #-----
35:
36: def main():
37:     if len(sys.argv) != 2:
38:         print('Usage: python %s port' % sys.argv[0])
39:         sys.exit(1)
40:
41:     try:
42:         port = int(sys.argv[1])
43:         server_sock = socket.socket()
44:         print('Opened server socket')
45:         if os.name != 'nt':
46:             server_sock.setsockopt(
47:                 socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
48:         server_sock.bind(('', port))
49:         server_sock.listen()
50:         while True:
51:             try:
52:                 sock, _ = server_sock.accept()
53:                 with sock:
54:                     print('Accepted connection')
55:                     handle_client(sock)
56:             except Exception as ex:
57:                 print(ex, file=sys.stderr)
58:         except Exception as ex:
59:             print(ex, file=sys.stderr)
60:             sys.exit(1)
61:
62: #-----
63: if __name__ == '__main__':
64:     main()

```

commpickle/client.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # client.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sys
9: import socket
10: import pickle
11:
12: #-----
13:
14: def write_courses(courses):
15:
16:     for course in courses:
17:         print(course['name'])
18:         book = course['book']
19:         print(book['title'])
20:         print(book['price'])
21:         print()
22:
23: #-----
24:
25: def main():
26:
27:     if len(sys.argv) != 3:
28:         print('Usage: python %s host port' % sys.argv[0])
29:         sys.exit(1)
30:
31:     try:
32:         host = sys.argv[1]
33:         port = int(sys.argv[2])
34:
35:         with socket.socket() as sock:
36:             sock.connect((host, port))
37:             with sock.makefile(mode='rb') as flo:
38:                 courses = pickle.load(flo)
39:
40:             write_courses(courses)
41:
42:     except Exception as ex:
43:         print(ex, file=sys.stderr)
44:         sys.exit(1)
45:
46: #-----
47:
48: if __name__ == '__main__':
49:     main()

```

commpickle/server.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # server.py
5: # Author: Bob Dondero
6: #-----
7:
8: import os
9: import sys
10: import socket
11: import pickle
12:
13: #-----
14:
15: def create_courses():
16:     # book0 = {'title': 'C Programming', 'price': 88.55}
17:     book1 = {'title': 'The Practice of Programming', 'price': 35.14}
18:     course0 = {'name': 'COS 217', 'book': book1} # Shared book.
19:     course1 = {'name': 'COS 333', 'book': book1} # Shared book.
20:     courses = [course0, course1]
21:     return courses
22:
23: #-----
24:
25: def handle_client(sock):
26:     courses = create_courses()
27:     with sock.makefile(mode='wb') as flo:
28:         pickle.dump(courses, flo)
29:         #flo.flush()
30:     print('Wrote to client')
31:
32: #-----
33:
34: def main():
35:     if len(sys.argv) != 2:
36:         print('Usage: python %s port' % sys.argv[0])
37:         sys.exit(1)
38:
39:     try:
40:         port = int(sys.argv[1])
41:         server_sock = socket.socket()
42:         print('Opened server socket')
43:         if os.name != 'nt':
44:             server_sock.setsockopt(
45:                 socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
46:         server_sock.bind(('', port))
47:         server_sock.listen()
48:         while True:
49:             try:
50:                 sock, _ = server_sock.accept()
51:                 with sock:
52:                     print('Accepted connection')
53:                     handle_client(sock)
54:             except Exception as ex:
55:                 print(ex, file=sys.stderr)
56:         except Exception as ex:
57:             print(ex, file=sys.stderr)
58:             sys.exit(1)
59:
60: #-----
61:
62: if __name__ == '__main__':
63:     main()

```