forking.py (Page 1 of 1)

```
 1: #!/usr/bin/env python
 2:
 3: #-------------------------------------------------------------------------
 4: # forking.py
 5: # Author: Bob Dondero
 6: #-------------------------------------------------------------------------
 7:
 8: import multiprocessing
 9:
10: def print_color(color):
11:     for _ in range(5):
12:         print(color)
13:     print(color + ' process terminated')
14:
15: def main():
16:
17:     blue_process = multiprocessing.Process(
18:         target=print_color, args=['blue'])
19:     red_process = multiprocessing.Process(
20:         target=print_color, args=['red'])
21:
22:     blue_process.start()
23:     red_process.start()
24:
25:     print('parent process terminated')
26:
27: if __name__ == '__main__':
28:     main()
```

waiting.py (Page 1 of 1)

```
 1: #!/usr/bin/env python
 2:
 3: #-------------------------------------------------------------------------
 4: # waiting.py
 5: # Author: Bob Dondero
 6: #-------------------------------------------------------------------------
 7:
 8: import multiprocessing
 9:
10: def print_color(color):
11:     for _ in range(5):
12:         print(color)
13:     print(color + ' process terminated')
14:
15: def main():
16:
17:     blue_process = multiprocessing.Process(
18:         target=print_color, args=['blue'])
19:     red_process = multiprocessing.Process(
20:         target=print_color, args=['red'])
21:
22:     blue_process.start()
23:     red_process.start()
24:
25:     blue_process.join()
26:     red_process.join()
27:
28:     print('parent process terminated')
29:
30: if __name__ == '__main__':
31:     main()
```

spawning.py (Page 1 of 1)

```
#!/usr/bin/env python

#-------------------------------------------------------------------------
# spawning.py
# Author: Bob Dondero
#-------------------------------------------------------------------------

import threading

#-------------------------------------------------------------------------

class PrinterThread (threading.Thread):

    def __init__(self, color):
        threading.Thread.__init__(self)
        self._color = color

    def run(self):
        for _ in range(5):
            print(self._color)
        print(self._color + ' thread terminated')

#-------------------------------------------------------------------------

def main():

    blue_thread = PrinterThread('blue')
    red_thread = PrinterThread('red')

    blue_thread.start()
    red_thread.start()

    print('main thread terminated')

#-------------------------------------------------------------------------

if __name__ == '__main__':
    main()
```


joining.py (Page 1 of 1)

```
#!/usr/bin/env python

#-------------------------------------------------------------------------
# joining.py
# Author: Bob Dondero
#-------------------------------------------------------------------------

import threading

#-------------------------------------------------------------------------

class PrinterThread (threading.Thread):

    def __init__(self, color):
        threading.Thread.__init__(self)
        self._color = color

    def run(self):
        for _ in range(5):
            print(self._color)
        print(self._color + ' thread terminated')

#-------------------------------------------------------------------------

def main():

    blue_thread = PrinterThread('blue')
    red_thread = PrinterThread('red')

    blue_thread.start()
    red_thread.start()

    blue_thread.join()
    red_thread.join()

    print('main thread terminated')

#-------------------------------------------------------------------------

if __name__ == '__main__':
    main()
```