std::future and std::promise Classes Solutions

Transfer of Data using Future and Promise

- Briefly describe how the producer-consumer model can be implemented using std::future and std::promise
  - An std::promise is associated with the producer
  - An std::future object is associated with the consumer
    - The consumer calls a member function of its future object
    - The function blocks until the result becomes available
  - The producer thread sends the result
    - Its promise object stores the result in the shared state
  - The consumer thread receives the result
    - The member function reads the result from the shared state
    - The member function returns the result

std::future

- What is the difference between the get() and wait() member functions of std::future?
  - Both member functions will block until the promise object stores the result in the shared state
  - The get() member function returns the result
  - The wait() member function returns nothing

std::promise Interface

- Which member function(s) does std::promise use to access the shared state?
  - set_value() to store the result
  - set_exception() to forward an exception

Producer-Consumer Model

- How would you create the std::promise and std::future objects, when writing a program that uses the Producer-Consumer model?
  - The parent thread creates the std::promise object by calling its constructor
  - It then calls the promise's get_future() member function, to obtain the associated future object
- How are these objects associated with the appropriate threads?
  - The std::promise is passed to the producer thread's task function, by reference
  - The std::future is passed to the consumer thread's task function, by reference