CSE 532 Spring 2020 Exam 2

Released 6:35am CDT Wednesday, April 15, 2020
Due via e-mail to eng-cse532@email.wustl.edu by
11:59pm CDT Thursday April 16, 2020

Name (please print clearly):

________________________________________

This exam is open-book and open-notes, and in general
covers the material since the first exam. Please answer each
question as completely and correctly as possible. Partial
credit will be given for incorrect answers that still show
understanding of the material.

During the exam you may use the any of the course text
books, lecture slides, and your labs, studios, notes and other
materials to which you have contributed during the
semester. You may also look up information on the internet,
such as the ACE library documentation, electronic copies of
the course text books, etc. However, communicating with
other people regarding the exam is not permitted – please
do your own research, formulate your own answers, and
prepare the exam document without interacting with others.

Please sign (or if completing the exam electronically, print)
your name below to indicate your understanding of and
compliance with the exam conditions noted above.

________________________________________
Exam scoring (to be completed by the grader)

<table>
<thead>
<tr>
<th>Question</th>
<th>Possible</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>8+1</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>100+1</td>
</tr>
</tbody>
</table>
1. (10 points) Please explain briefly why the `handle_input` method of an event handler that is registered with an `ACE_Reactor` should only make a single call to the `recv` method of an `ACE_SOCK_Stream` before returning.

What method of an event handler that is registered with an `ACE_Reactor` using `ACE_Event_Handler::WRITE_MASK` is called when there is sufficient room for a packet in the corresponding TCP/IP socket’s outbound buffer?

What method of an event handler that is registered with an `ACE_Reactor` for `SIGINT` is called when the user hits `Ctrl-C` in the terminal window where it is running?

What method of an event handler that is registered with an `ACE_Reactor` via the reactor’s `schedule_timer` method is then called repeatedly (once per the specified interval)?

What method of an event handler that is registered with an `ACE_Reactor` is called when that handler is removed from the reactor or when the reactor is shut down?
2. (12 points) Please describe briefly two different example scenarios in which the Interceptor pattern can be applied effectively:

Please describe (again briefly) how the two applications of the Interceptor pattern are similar:

Please describe (again briefly) how the two applications of the Interceptor pattern differ:

Please describe briefly how applying the Extension Interface pattern along with Interceptor could help in the design of one (or both) of the examples you gave, along with any other consequences that applying the Extension Interface pattern would have in that context:
3. (12 points) Please briefly describe a situation in which the Leader/Followers pattern is likely to offer better performance than the Half-Sync/Half-Async pattern:

Please explain how the behaviors of the threads involved will differ when the Leader/Followers pattern is applied, versus when the Half-Sync/Half-Async pattern is applied:

Please explain why these differences in thread behavior may result in better performance with Leader/Followers rather than Half-Sync/Half-Async.
4. (12 points) Please describe briefly the intent of the Component Configurator pattern:

Please give an example of a situation in which it would be useful to apply the Component Configurator pattern and please explain briefly why it would be useful to do so:

Please describe briefly how with that pattern the system could be suspended and resumed safely and correctly:

Please name two or more system features that if present could make it undesirable to apply the Component Configurator pattern, and explain briefly why that is so:
5. (10 points) Next to each term please write the letter corresponding to the definition that best matches it (and vice versa).

a. Which library to link with ___ -I g++ option
b. File produced by building ACE ___ -L g++ option
c. Enables newest features ___ -l g++ option
d. Root directory for ACE ___ -std=c++17 g++ option
e. Removes binary/temp files ___ g++ error
f. An environment variable ___ g++ warning
g. Where to find header files ___ ACE_Wrappers
h. Where to find library files ___ ACE_ROOT
i. Lets compilation finish ___ make clean
j. Stops compilation ___ libACE.so

6. (4 points) Please put each of the following words into the appropriate underscored place in the following sentence: acceptor, connector, handler, reactor:

A/an ___________ demultiplexes events onto a/an event ___________. A/an ___________ initiates connection establishment requests, while a/an ___________ listens for connection establishment requests.
7. (8 points) Please name a concurrency pattern that was used in your lab2 or lab3 solution:

Please describe briefly how it was used in that lab solution:

Please name a synchronization pattern that was used in your lab2 or lab3 solution:

Please describe briefly how it was used in that lab solution:

Extra credit (1 point): Please fill in the missing word at the end of the following sentence, in the underscored place indicated:

The ACE_Event_Handler::get_handle method is an example of a controlled violation of ______________.
8. (12 points) Please explain briefly why it may be important to have separate event handler classes for accepting connections versus for socket input and output, in a server that needs to interact with an unknown number of remote client programs:

If the event handler that accepts connections dynamically allocates objects of the other event handler class whenever a new socket connection is established, and transfers ownership of the connection to that newly created event handler, what two (or more) things must happen when the newly established socket connection has been used and is then closed by a client?

Please explain briefly how each of those things can be managed entirely within the dynamically allocated event handler object, independent of the main server program and of the event handler that created it:
9. (8 points) Please name and describe briefly two kinds of unwanted blocking that may occur in a multi-threaded program:

Please name and describe briefly two kinds of broken invariants that can occur in a multi-threaded program:

Please explain briefly how recording events (such as when attempts to acquire or release a mutex are made, within which functions/methods, and by which threads) within a multi-threaded program can help to narrow down where a problem (like a deadlock) is occurring:

Please explain briefly why a multi-threaded program may be significantly more complex to debug, compared to a single-threaded version of the same program:
10. (12 points) Please explain briefly what is meant by the vertical design of a multi-threaded system architecture:

Please explain briefly what is meant by the horizontal design of a (distributed) multi-threaded system architecture:

Please explain briefly how deadlock could arise when using the WaitOnConnection strategy in a distributed multi-threaded system architecture:

Please describe briefly a solution approach for how deadlock can be avoided when using the WaitOnConnection strategy in a distributed multi-threaded system architecture (without using a stackless approach).