This exam is a closed book, closed notes, closed laptops and smartphones, etc. exam
All that you can use is either a pen or pencil
The last few pages provide formula, which you may be helpful
Do not spend too much time on any one question.

Honor Code statement: I pledge that this submission is solely my work. I pledge that I have not provided help to anyone. I pledge that I have not received help from anyone.

Signature

<table>
<thead>
<tr>
<th>Section</th>
<th>Question Type</th>
<th>Question Number(s)</th>
<th>Points Possible</th>
<th>Points Earned</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>True/False</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>II</td>
<td>Multiple Choice</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>III</td>
<td>Semaphores</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IV</td>
<td>Short Answer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>Multiple Answers</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total without curve 100

Curve Adjustment

| VI      | Extra Credit  | One or more |                  |

Total Exam Score
I. True/False. Instructions: Circle either True or False. No partial credit. 3 points each.

1. True / False  A lightweight process shares a stack with other processes, but has its own heap

2. True / False  Necessary delay is a condition that is imposed by a correct solution to the critical section problem

3. True / False  `<await (C);>` can be implemented via `for(i=0; i<1; i--){}`

4. True / False  Whether the final state of a program or system is correct is a liveness Property

5. True / False  Flow dependence is when an instruction depends on the results of a previous instruction

6. True / False  The Test and Set instruction echoes (returns) a Boolean value that is equal to the value of the Boolean variable that the instruction receives as the only argument
II. Multiple Choice

Instructions: Circle the ONE letter to specify the BEST answer choice from among those listed. No partial credit.

7. Assume that a semaphore, aSema, is initialized with a value of 2. Which of the following statements is true.

A. Decrementing aSema (from 2 to 1) will cause the process that issued the decrement to self-block
B. aSema implements a multiplex, which specifies the number of concurrent threads that are allowed to have access to a shared resource
C. Incrementing aSema will cause the process that issued the increment to self-block
D. Assume that there are two processes, P1 and P2, that each issue aSema.decrement(). A third process, P3, issues aSema.decrement(), followed by process P4 which also issues aSema.decrement(). If then P5 issues aSema.increment(), then both P4 and P3 are unblocked.
E. None of the above

8. If the below for loop were parallelized, how many different ways could the parallel version of the program print out the values of i? Assume print(i) is an atomic action.

```plaintext
for (i=5; i<15; i+=2){
    print(i);
}
```

A. 1
B. 3
C. 10
D. 15
E. 16
F. 81
G. 120
H. None of the above
III. Semaphores. Partial credit given.

9. The pseudocode below contains three functions, calcA() and calcB(), and main(). All instructions inside calcA and calcB are atomic. Assume that a Semaphore class is available, which has a constructor Semaphore(int semaphoreValue), and functions increment() and decrement(). Variables x, y and z are global variables (ie, shared among all functions) saved in shared memory. Declare and use semaphores wherever needed, so that when the main method runs to completion, the output of the program is:

3, 0, 6

You can ONLY declare and use semaphores. You cannot update existing variables’ values, insert new code other than semaphores, etc.

```c
function calcA(){
    x = x + 1;

    y = y - x;
}

function calcB(){
    z = 2 * x;
}

main(){
    x = 2; y = 3; z = 4;
    co

    calcA();

    calcB();

    oc
    print(x, "", y, "", z);
}
```
V. Short Answer. Provide a concise answer to each question. Partial credit.

10. Among 10 processes (workers), what is the minimum number of 3-process barriers (among all stages) that are needed to enable all 10 processes to synchronize with each other?

11. Assume a 3 step pipeline with the following stage latencies

<table>
<thead>
<tr>
<th>Stage</th>
<th>Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage_1</td>
<td>2ns</td>
</tr>
<tr>
<td>Stage_2</td>
<td>3ns</td>
</tr>
<tr>
<td>Stage_3</td>
<td>1ns</td>
</tr>
</tbody>
</table>

The pipeline is not superscalar. Three instructions, i1, i2, and i3, are queued and ready to be executed via the pipeline. Assuming that the pipeline is originally empty, how much time is needed for the pipeline to process all three instructions, and what is the total number of stalls experienced by the 3 instructions? Assume that an instruction waiting to ENTER the pipeline does not count as a stall. Only after an instruction has entered the pipeline, if it is delayed, does it count as a stall.

Time needed to process the 3 instructions: _________________________________

Total number of stalls experienced by the 3 instructions: ____________________

12. What cache reuse ratio is needed so that a program that uses cache is 3 times better (performance gain) than a program that does not use cache? Assume cache access time of 5ns and memory access time of 25ns.

Your Answer: _________________________________
13. There are 60 processes, each of which has a locally scoped integer variable. The processes communicate via MPI. If the goal is for each process to learn the smallest and largest locally scoped variable among the entire 60 processes, how many more messages will a symmetric topology network require than a centralized one? Assume that network connections among processes are stable, and that the uptime for all 60 processes is 100%.

Your Answer: ___________________________________________________________________

14. Assume the below code statements S1 and S2, and that x and y are both initially 7. Both x and y are shared variables.

S1: x = y + 1;
S2: y = x - y;

Under these conditions what are the possible final values of x and y after P1 has been allowed 12 hours to execute? Assume there are no other processes running on the computer where P1 is being executed.

P1: co <await(y<7)> <S2>; // <S1>; oc

Possible value(s) of x: ___________________________________________________________________

Possible value(s) of y: ___________________________________________________________________

15. Explain why the code on the right is more efficient than the code on the left, when n is large.

```java
while(true){
    co [i = 1 to n]
        performCalculation(i);
    oc
}
```

Process CS[i = 1 to n]{
    while(true) {
        performCalculation(i);
        barrier common to all
    }
}

Your Answer: ___________________________________________________________________

___________________________________________________________________________

___________________________________________________________________________

16. Explain what it means for an instruction to be executed atomically

Your Answer: ___________________________________________________________________

___________________________________________________________________________

___________________________________________________________________________
VI. Multiple Answers. Select ALL letters that are correct answers for the question. One, two or more, or even all choices may constitute a fully correct answer. Partial credit is given.

17. Which of the following is/are **true** about monitors?

   A. Access to monitor variables by an outside object is direct because monitor variables are declared public
   B. Monitors are active processes because they interact with and make changes to external objects
   C. A monitor’s state, as specified via its condition variables, is not directly accessible from anything that is outside of the monitor
   D. Only 1 process can be executing any one statement in any procedure in a monitor at any one time.

VII. Extra credit. Partial credit not given.

   Two extra credit questions will appear on the real exam.

VIII. Formula, Axioms, etc.

   **Cache/Main memory gain**

   \[ G(\tau,\beta) = \frac{\tau}{\beta + \tau(1-\beta)} \]

   \[ T_c = \frac{T_m}{\tau} \]