

I. Concurrent Programming Language Constructs:

A. Creating multiple processes:

i. Fork / Join

ii. Co-begin / Co-end (also known as **ParBegin** and **ParEnd**)

B. Hardware Instructions:

i. Test-And-Set

ii. Swap (atomic)

C. Semaphores

i. Wait

ii. Signal

D. Mutexes (ie, how are they different than semaphores?)

E. Critical Regions (not the same as critical sections)

F. Monitors

II. Algorithms:

A. Mutual Exclusion:

i. Using Swap

ii. Using Test-And-Set

iii. Bakery Algorithm

iv. This unnamed algorithm:

```
// note: i = 1 - j;

while(1)
{
    flag[i] = true;
    turn = j;
    while(flag[j] && turn = j)
        no-op;
    // Critical section

    // Exiting critical section
    flag[i] = false;
    // Remainder section
}
```

B. Semaphore Examples:

i. Dining Philosophers

ii. Cookie Jar Problem

iii. Reader-Writer Problem

iv.Unisex Bathroom

III.Terms and Concepts (definitions):

A.Invariant Properties

B.Safety Conditions

C.Safety Conditions for Critical Regions:

i. Mutual Exclusion

ii. Progress

iii.Bounded waiting

D.Liveness Conditions

E.Concurrent Program

F.Sequential Program