King Saud University

College of Computer and Information Sciences
Computer Science Department

Course Code:	CSC 227
Course Title:	Operating Systems
Semester:	Summer 2015
Exercises Cover Sheet:	**Mid 2 Exam**

Duration: 90 min

Student Name:	
Student ID:	
Student Section No.	

Tick the Relevant	**Computer Science B.Sc. Program ABET Student Outcomes**	**Question No. Relevant Is Hyperlinked**	**Covering %**
	a) Apply knowledge of computing and mathematics appropriate to the discipline;		
	b) Analyze a problem, and identify and define the computing requirements appropriate to its solution		
	c) Design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;		
	d) Function effectively on teams to accomplish a common goal;		
	e) Understanding of professional, ethical, legal, security, and social issues and responsibilities;		
	f) Communicate effectively with a range of audiences;		
	g) Analyze the local and global impact of computing on individuals, organizations and society;		
	h) Recognition of the need for, and an ability to engage in, continuing professional development;		
	i) Use current techniques, skills, and tools necessary for computing practices.		
	j) Apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices;		
	k) Apply design and development principles in the construction of software systems of varying complexity;		

Question 1. [12 marks] Select ONLY ONE ANSWER (the best answer).

Copy your answer for question 1-1 to 1-15 in the table on page2. ONLY THAT TABLE WILL BE GRADED.

1	Which one of the following is not shared by threads?		2	Termination of the process terminates
a	program counter		a	first thread of the process
b	Stack		b	first two threads of the process
c	both (a) and (b)		c	all threads within the process
d	none of the mentioned		d	no thread within the process
3	The register context and stacks of a thread are deallocated when the thread		4	Instead of starting a new thread for every task to execute concurrently, the task can be passed to a ___________.
a	terminates		a	thread pool
b	Blocks		b	process
c	Unblocks		c	thread queue
d	Spawns		d	None of these
5	Thread pools help in :		6	An un-interruptible unit is known as :
a	servicing a single request using multiple threads from the pool		a	static
b	servicing multiple requests using one thread		b	Single
c	faster servicing of requests with an existing thread rather than waiting to create a new thread		c	Atomic
d	None of these		d	None of these
7	____________ occurs when a higher-priority process needs a resource that is currently being accessed by a lower-priority process.		8	A solution to the critical section problem must satisfy
a	Priority inversion		a	Mutual Exclusion
b	Deadlock		b	Progress
c	A race condition		c	Bounded waiting
d	A critical section		d	All of the above
9	-------- occurs when a process has to loop continuously before it can enter its critical section while another process is in its critical section		10	Semaphores **cannot** be used for
a	Race condition		a	Managing criticial sections (i.e. mutual exclusion)
b	Busy waiting		b	Controlling access to a given resource consisting of a finite number of instances
c	Non-preemptive scheduling		c	Synchronizing the execution of sentences
d	Deadlock		d	None of the above

11	If a programmer misused semaphores by using first signal() then wait(), then		12	If a programmer misused semaphores by using first wait() then wait()
a	Deadlock would occur		a	Deadlock would occur
b	Starvation would occur		b	Starvation would occur
c	The mutual-exclusion requirement would be violated		c	The mutual-exclusion requirement would be violated
d	All of the above		d	All of the above

1.	2.	3.	4.	5.	6.	7.	8.	9.	10.

11.	12.

Question 2 (3 marks)

Mark each of the following statements with either T (for True statements) or F (for false statements).

1. Preemptive kernels are more responsive than non-preemptive kernels.
2. The test_and_set instruction is a hardware atomic instruction
3. The semaphore operation wait() and signal atomic instructions
4. We say that starvation occurred when every process in a set of processes is waiting for an event that can be caused only by another process in the set.
5. In the dining philosophers problem deadlock would occur if each philosopher picked up the right chopstick first.
6. Data are share between threads in Java using global variables

Question 3[5 marks]

2-a) [1 mark] What are the two main approaches used for thread cancellation

....

....

....

2-b) [1 mark] In what way are user-level threads better than the kernel-level threads.

....

....

2-c) [1+1 mark] What are the differences between user-level threads and kernel-supported threads?

....

....

Question 4

A) (3 marks) Write a pseudo code for the test_and_set operation that is used by modern machines for locking purposes

....

....

....

....

....

....

....

B) (4 marks) Write a pseudo code that describes how to use test_and_set operation to resolve the critical section problem.

....

....

....

....

....

....

....

Question 5

I) Consider the following code of a producer process that uses semaphores to solve the bounded buffer problem

```
do{                                                                 ...
   /* produce an item in next_produced */
     ...
   wait(empty);
   wait(mutex);
     ...
   /* add next produced to the buffer */
     ...
   signal(mutex);
   signal(full);
 } while (true);
```

a) (3 marks) What are the initial values for each of the following semaphores as used to solve the bounded buffer problem

1) empty

....

2) mutex

....

3) full

....

b) (4 marks) Explain the purpose of using each of these statements in the code

1) wait(empty)

....

....

2) wait(mutex)

....

....

3) signal(mutex)

....

....

4) signal(full)

....

....

II) (2 marks) Consider the following code for processed P_0 and P_1, where S and Q be two semaphores initialized to 1,

```
      P0                  P1
  wait(S);            wait(Q);
  wait(Q);            wait(S);
   ...                  ...
  signal(S);          signal(Q);
  signal(Q);          signal(S);
```

What would happen if the following order of execution took place and explain why

1) P0 executes wait(S)
2) P1 executes wait(Q)
3) P0 executes wait(Q)
4) P1 executes wait(S)

....

....

....

....

Question 6

Consider the following Java code

```
class Sum
{
   private int sum;

   public int getSum() {
     return sum;
   }

   public void setSum(int sum) {
     this.sum = sum;
   }
}

class Summation implements Runnable
{
   private int upper;
   private Sum sumValue;

   public Summation(int upper, Sum sumValue) {
     this.upper = upper;
     this.sumValue = sumValue;
   }

   public void run() {
     int sum = 0;
     for (int i = 0; i <= upper; i++)
        sum += i;
     sumValue.setSum(sum);
   }
}
```

1) (2 marks) What is the purpose of implementing the interface runnable by class Summation?

....

....

2) (3 marks) Write the necessary Java statement that allow us to use method run() in class summation

....

....

....

....