**BRAC UNIVERSITY**

**Department of Computer Science and Engineering**

|  |  |
| --- | --- |
| Examination: Final  Duration: 2 Hours | Semester: Spring 2022  Full Marks: 40 |

CSE 321: Operating Systems

Answer the following questions.

Figures in the right margin indicate marks.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1.  **CO5** | 1. **Explain** Race conditions with an example. **Mention** how we protect the system from this phenomenon. 2. **Explain** with a code example how a careless ordering of semaphore operations can lead to a deadlock situation among two processes. 3. Suppose, you have to design an online consultation system for the teachers and students of your university. There are certain constraints that you have to keep in mind while designing the system -   i. the teachers can set their status whether they are available to give consultation or not. A teacher will set him unavailable after the giveConsultation() function.  ii. the students will enter a voice channel if the teacher is available for consultation.  iii. the students will wait if one student is in consultation with the teacher  Now, you have to **design** the teacher and student function using semaphores so that synchronization can be achieved among them maintaining the constraints mentioned above. You can use the following code template given below and complete it. **Mention the initial semaphore values before writing the functions.**   |  | | --- | | //initialize the semaphore values here  S=1  teacher(){  // write semaphore code here  *giveConsultation(s);*      // write semaphore code here  }  student(){  // write semaphore code here  *takeConsultation(s);*  // write semaphore code here  }  Take consultation(s,std):  s-=1  if s<0:  std.append(list)  else:  give consultation(s, std) | | Give consultation(s,std):  S++  If s>=0:  Wakeup(std) | | [2+1]  [2]  [5] |
|  |  |  |
|  |  | [3]  [2]  [4]  [2]  [4] |
|  |  |  |