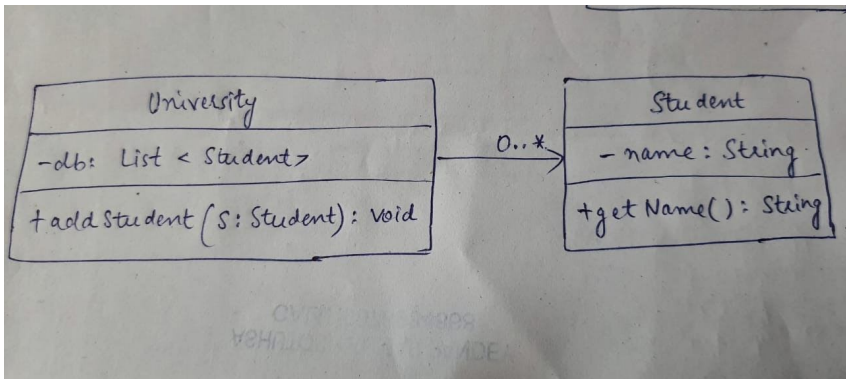
Advanced Programming (CSE201), Quiz -3

Time allocated: 04:15pm – 4:35pm (20 minutes)

Instructions:

- This is a closed book quiz and duration is 20 minutes. We are giving another 10 minutes (4.35PM - 4.45PM) for submission in all the modes.
- Only reasonable and clearly mentioned assumptions (if any) would be accepted.
- For justifications, please be as concise as possible (2-3 sentences only)
- **IIIT plagiarism policy is applicable if any such cases found**
- **Write your answers on a plain sheet that you can upload by taking a picture of the same (ensure low resolution so that the upload size is smaller)**
- **You can either submit your solutions on the below Google Form link https://docs.google.com/forms/d/e/1FAIpQLSe_zh5mShPMaPPjczj69pnCavmNXh7Xn-XYXaBt8yZDOlU9xA/viewform?usp=sf_link or email the quiz solutions to "ap-m2020-submission@iiitd.ac.in" or upload on google classroom . Only one mode of submission!**
- **Subject of the mail should be the Quiz-3 and this naming convention to be followed to save your submission: Rollno_Quiz3.**
- **We will not consider any submission that is emailed beyond 4.45pm. It is your responsibility to ensure you email it or submit through google form or through google classroom on time. Please ensure you have proper internet connectivity as we are giving you sufficient extra time.**

Question-1: Represent the UML diagram shown in Java code [**2 marks]**



Answer:

```
class Student {
private String name;   [+0.5 marks]
public String getName() {  [+0.5 marks]
 return name; }
}
```

```
class University {
private List<Student> db = new ArrayList<Student>();  [+0.5 marks]
public void addStudent(Student s) {  [+0.5 marks]
db.add(s); }
}
```

Question-2: Arrange the following events in chronological order in a multithreading execution (just mention the ordering in terms of letters A, B, C, D and E). **[2.5 marks]**

A. Execution of "return" statement in "public void run()" method
B. Calling "start()" method in Thread class
C. Calling "new Thread(.....)"
D. Unblocking of "join()" method in Thread class
E. Calling "sleep()" method in the spawned Thread

Answer:

C, B, E, A, D
(All these 3 options must be shown and in SAME order). [+2.5 marks] //No partial marking

Question-3: Arrange the following events in chronological order (just mention the ordering in terms of letters A, B, C, D, E and F) representing the execution of a parallel program by using a thread pool. For ensuring the correct execution you are allowed to mention any of the following events more than once in the ordering that you mention below **[2.5 marks].**

A. Creation of ForkJoinPool
B. SubTask.join()
C. Creation of RootTask
D. Execution of method "public void compute()"
E. SubTask.fork()
F. ForkJoinPool.invoke(RootTask)

Answer:

"A, C, F, D, E, D, B" (OR) "C, A, F, D, E, D, B"
(Some students might have shown the last 3 letters "E,D,B" several times and that is also correct. E.g.: "A,C,F,D,E,D,B,E,D,B,E,D,B,E,D,B……")
[+2.5 marks] //No partial marking

Question-4: Two threads T1 and T2 are created by using the **same** Runnable type object of class "MyClass". What will be the execution time of the below program? Can you improve its

execution? How? What would be the execution time with your updated code? Assume you are running this program on a hypothetical multicore system that has lots of free cores. Assume "fibonacci" is a sequential method to calculate a fibonacci number. **[3 marks]**

```
class MyClass implements Runnable {
   public Integer result;
   public MyClass (Integer r) { result=r; }
   public synchronized void run() {
      int N=40;
      if(Thread.getName().equals("T1") {
         result += fibonacci(39); //=4sec
      } else {
         result += fibonacci(38); //=3sec
      }
   }
   ......
}
```

Answer:

Execution time: 7 seconds. [+0.5 marks]
Improved code:

```
public void run() {    [+0.25 marks]
int N=40;
int a, b;
if(Thread.getName().equals("T1")) {    [+0.25 marks]
a = fibonacci(39);    [+0.25 marks]
} else {
b = fibonacci(38);    [+0.25 marks]
}
synchronized(this) {    [+0.5 marks]
result = a + b;              [+0.5 marks]
}
}
```

Execution time of updated code: 4 seconds [+0.5 marks]