Given the following relations:

- enrolled(snum:integer,cname:string)
- class(cname:string,meets_at:time,room:string,fid:integer)
- faculty(fid:integer,fname:string,dept:string)
- student(snum:integer,sname:string,major:string,level:string,age:integer)

Provide SQL instructions for each of the following questions

1. (a) [**8 points**] Create the database schema, including appropriate versions of all primary and foreign key integrity constraints. If you need assumptions, write them down.

   (b) [**2 points**] If you have to populate the tables with outside data. What is the correct order in which the tables must be filled?

2. [**5 points**] Find the names of all professors that work for the 'cs' department and whose name starts with 'W'.

3. [**5 points**] List of majors (no duplicates) of all students enrolled in the 'Algorithms' class.

4. [**5 points**] Names of all students that have the same age as 'Horatio' (assume that there is only one student called Horatio).

5. [**5 points**] Names of all students older than all students whose name starts with an 'A'.

6. [**15 points**] USE CORRELATED SUBQUERIES to find the name of those courses that have no students (no credit given if you don't use correlated subqueries).

7. Use set operations to

   (a) [**10 points**] Determine the names of professors who teach cs and math majors.

   (b) [**5 points**] Determine the names of professors who teach at least a class that has both math and cs majors.

   (c) [**5 points**] determine the names of professors who teach cs majors and not math majors.

8. (a) [**3 points**] Find the average age of all students.

   (b) [**7 points**] Find the names of all students that are older than average.

9. [**10 points**] For each faculty member (name) determine the average age of all students enrolled in his/her classes.

10. [**15 points**] Use outer join to determine the number of classes that each student is enrolled in. Your result must include all students, even if they are not taking any classes.