Assignment -1
CS 308: Software Engineering

1. Define the following terms precisely
   Software, Software Engineering, Technique, Model, Tool, validation, SDLC, software process.

2. How programmer's productivity is typically measured? Given that an online registration software is developed by a team of four student developers in three chronological months. The software consisted of 4000 lines of non-commented Java code. Compute the programmer's productivity in this project.

3. Software cost can broadly categorized into two types: Development cost and Maintenance cost. The maintenance cost is considered to be of same magnitude of the development cost. Of the development cost, an example distribution of effort with the different phases could be:

Requirement	10%
Design	10%
Coding	30%
Testing	50%

   It is reasonable to assume that if software is easy to test, it will be easy to maintain. Suppose that by putting extra effort in design and coding you increase the cost of these phases by 20%, but you reduce the cost of testing and maintenance cost by 5%. Will you put in the extra effort?

4. In a study, it has been found that the approximate average cost of fixing requirement errors (in person-hours) depending on the phase is shown below

Phase	**Cost (person-hour)**
Requirement	2
Design	5
Coding	15
Testing	50
Maintenance	150

   Consider that an error that remains in the requirements will be detected in the later phases with the following probabilities: 0.4 in design, 0.1 in coding, 0.4 in testing, and 0.1 in maintenance. Assume that there are 50 requirement errors that get into the system during requirement phase. Find

   (a) What will be the average effort required to identify and fix the 50 requirement errors if not detected in the requirement phase itself.

   (b) What will be saving in the efforts (in theory) applied if all the 50 requirement bugs got detected in the requirement phase itself?

   (c) The answer of (b) may not be applicable in practice. Give reasons why?

5. What are the different attributes of software quality? If for an accounting software we are most interested in ensuring that the software does not make any computation mistakes, then which of the quality attributes should we be most concerned about?

6. What are some of the project management tasks that you will do differently for a large project as compared to a small project? How will your execution of these tasks change?

7. Suppose changes are to be made to a software system that is in operation. Why will changes to such a system cost a lot more than just making changes to the source code files?