Additional sample questions for the midterm (see also the exercises in class, lecture notes, tutorials)

Question 1

1.1 Consider the following measurement results: (CC is cyclomatic complexity; EC is essential complexity)

Name of module	CC	EC
Module_1	**10**	**9**
Module_2	**8**	**9**
Module_3	**10**	**1**

a) Which module is completely structured?

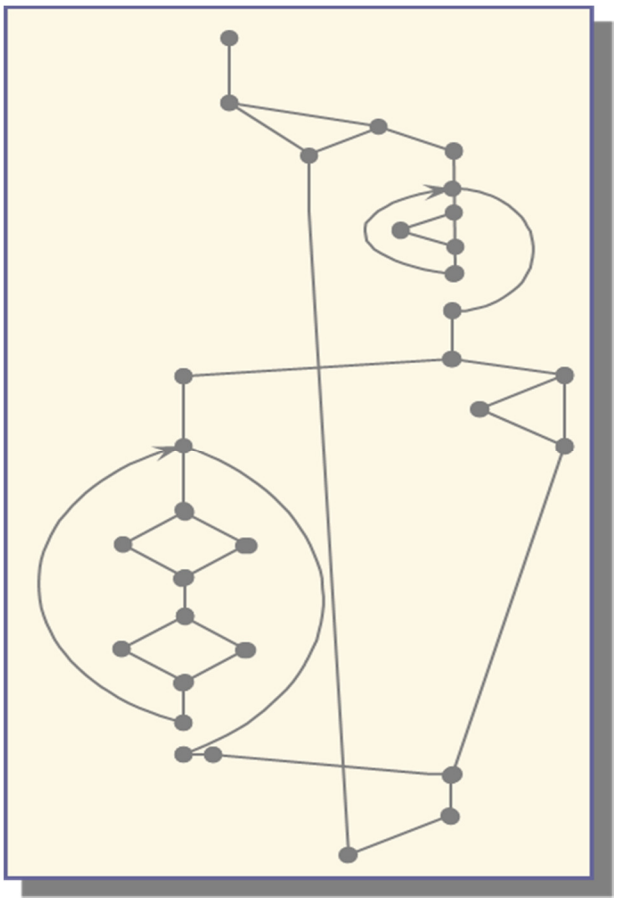
Module_1 Yes / No
Module_2 Yes / No
Module_3 Yes / No

b) Which module needs more testing?

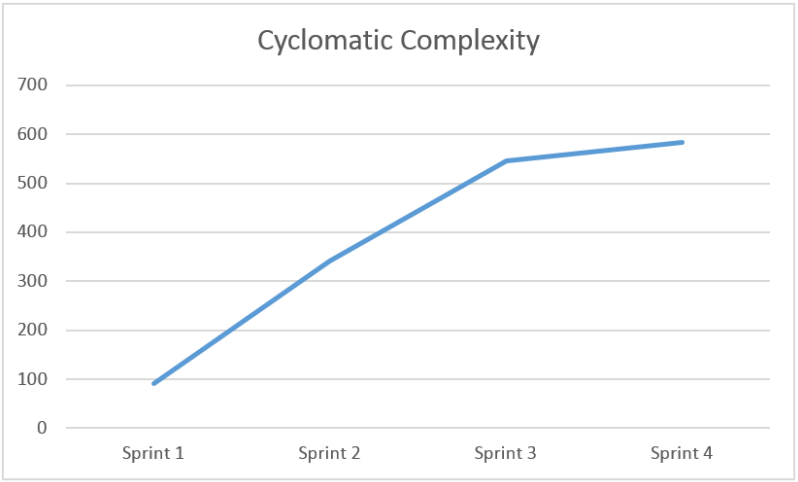
Module_1 Yes / No
Module_2 Yes / No
Module_3 Yes / No

1.2 The number of linearly independent paths of the graph is :

a) 8
b) 10
c) 3
d) 11
e) None of the above



1.3 Code quality trend depict graphically the trends of the chosen metrics. Interpret the Cyclomatic Complexity results below to attempt to reason why certain trends decreased or increased



Hint: see Lecture 5, slide 28

Question 2

Explain briefly why measurement feedback is important in software management?

Question 3

Suppose you as a developer have a goal to "deliver defect free versions". Use the Goal-Question-Metric approach to suggest two relevant questions, their corresponding indicators and the base measurements that will enable you to determine if you've met your goal.
Hint: see Lecture 5 slides 19 to 25 and the measurement program discussed in lecture 5

Question 4

List the eight ISO 25010 quality characteristics and write there definitions.

Question 5

A. Draw an OO quality Factor-Criteria-Metric model based on the measurements studied in this course (Hint: see Lecture 12, slide 52 (was 26))

B. Compute CK measures Number of Children (NOC) and Depth of Inheritance Tree (DIT) for the classes in the Robotics hierarchy shown below. Interpret the results.

```
class Robot {
public int x, y;

    public void up() { y=y+1;}
    public void down () {y=y-1;}
    public void right () {x=x+1;}
    public void left () {x=x-1;}
    public void up_right () {x=x+1; y=y+1;}
    public void down_right () {x=x+1; y=y-1;}
    public void up_left () {x=x-1; y=y+1;}
    public void down_left () {x=x-1; y=y-1;}
}

class Robot_sense extends Robot {
     public double distance () {
     double xdiff = x - Target.x;
     double ydiff = y - Target.y;
     return Math.sqrt(xdiff*xdiff + ydiff*ydiff)
          }
}
```

```
class Robot_sense_warn extends Robot_sense {
     public Boolean warn () {
     double xdiff = x - Target.x;
     double ydiff = y - Target.y;
     return { if (xdiff <a && ydiff <a) warn = True else warn = False}
}
```

Question 6

1) Which of these are reasons for using measures during software development?

   A. large body of scientific evidence supports their use
   B. provides software engineers with an objective mechanism for assessing software quality
   C. they allow all software quality information to be expressed unambiguously as a single number

2) Software measurement program set up for an organization is driven by the business or technical goals an organization wishes to accomplish.

   A. True
   B. False

3) Determine which of the following statements is meaningful:

   A. The length of Program P is 10 KLOC.
   B. The arithmetic mean of the maintainability level of Programs P,Q, R and S is 4.75
   C. Program A took 10 person-months to develop

4) Cohesion is an indicator of the degree to which a module
   a. can be written more compactly
   b. focuses on just one thing
   c. is able to complete its function in a timely manner
   d. is connected to other modules and the outside world

5) Coupling is an indicator of the degree to which a module
   a. can be written more compactly
   b. focuses on just one thing
   c. is able to complete its function in a timely manner
   d. is connected to other modules and the outside world

6) The depth of inheritance tree (DIT) metric can give an OO software designer a reading on the
   a. attributes required for each class
   b. completion time required for system implementation
   c. complexity of the class hierarchy
   d. level of object reusability achieved

Question 7

For each one of class measures given in the table below, write the attribute it measures:

MEASURE	SOFTWARE ENTITY	ATTRIBUTE
Depth of Inheritance Tree DIT	*Class*	*Inheritance*
Coupling Between Objects CBO	?	?
Response For Class RFC	?	?
Number Of Children NOC	?	?
Weighted Methods per Class WMC	?	?

Question 8

Company X runs a large software system S that has been developed in-house over a number of years. The company collects information about software defects discovered by users of S. During the regular maintenance cycle, each defect is traced to one of the nine subsystems of S (labeled with letters A through I), each of which is the responsibility of a different team of programmers. The table below summarizes information about new defects discovered during the current year.

System	A	B	C	D	E	F	G	H	I
Defects	35	0	95	35	55	40	55	40	45
Size (KLOC)	40	100	5	50	120	70	60	100	40

1. Compute the defect density for each subsystem
2. Use box-plot outlier analysis to identify unusual features of the system
3. What conclusions can you draw from your outlier analysis?