Q1	20
Q2	10
Q3	6
Q4	5
Q5	20
Q6	10
Q7	5
Q8	15
Q9	9
Total	100

Name:_____ CS3733 Final Exam 99D

1. [**20 pts**] Preventative maintenance is any effort to correct the structure of a software system to reduce the costs of future maintenance. Explain the role (if any) that preventative maintenance plays in the following life cycles:

(a) Waterfall Model

(b) Incremental Model

(c) Spiral Model

(d) Recursive Parallel Model

2. [**10 pts.**]

(a) What is the reason for creating a software prototype that will be thrown away once completed?

(b) What is the reason for creating a software prototype that will evolve over time into a complete, working system?

3. [**6 pts**] Identify three distinct reason for developing comprehensive use case scenarios.

4. [**5 pts**] Which best describes a component and why: (a) a set of interfaces; (b) a set of classes and objects.

5. [**20 pts**]

(a) Explain an impact that low cohesion has on a class.

(b) Evaluate the cohesion of the Main class that each group implemented for AMS.

(c) Does the concept of cohesion apply to an interface? If so, explain how.

(d) Explain why inheritance is a form of coupling between a class and its superclass.

6. [**10 pts**] Describe the difference between top-down implementation and. bottom-up implementation as it applies to Object-Oriented technology.

7. [**5 pts**] Why does top-down implementation often lead to reusable classes that are poorly tested?

8. [**15 pts**] True or false?

Black-Box Testing requires the specification of the software being tested	True	False
White-Box Testing requires the pseudo-code of the software being tested	True	False
In Object-Oriented testing, the object is the smallest unit of testing	True	False
A superclass inherits the protected and public methods of its subclass	True	False
A private method can only be invoked by the class in which it is defined	True	False

9. [**9 pts**]

(a) A method for class **C** is designed to throw an exception to signal an error. How does this affect the coupling between the calling object and **C**?

(b) Explain why the `java.util.Enumeration` class is an example of iterative abstraction.