Unit 1 Questions (Same groups as your Unit 1 project)

Due: 11th February, 5:00 pm

Please answer the following questions. Some may be more open ended than others. Diagrams are highly recommended when answering some of these questions (please use appropriate UML notation).

1. Does pair programming reduce the need for refactoring? Support your argument with appropriate evidence.

2. In "No Silver Bullet -- Essence and Accidents of Software Engineering" by Fred Brooks, the author claims "there is no single development, in either technology or management technique, which by itself promises even one order of magnitude improvement within a decade in productivity, in reliability, in simplicity". List arguments to support/contradict his claim.

3. The C++ implementation of class Adapter specifies the use of multiple inheritance. Using a diagram show how this could be implemented in Java? Be specific about the Java features that you are using.

4. One method of classifying iterators does so along two dimensions. The first indicates the location of control of the iteration (internal to the iterator or external client control), and the second which indicates the location of the definition of the iteration logic (embedded as part of the collection objects or in objects separate from the collections). Considering each dimension separately, what are the positive and/or negative aspects of iterators of each type?

5. What language constructs would you use to give iterators privileged access in Java and in C++? How will your answer depend on the classifications for iterators given above?

6. Draw sequence diagrams for the registration and update cycles of an Observer pattern implemented using appropriate Java classes. Be sure to label the object with the Java class and its role in the Observer pattern using Stereotypes.

7. You have seen that it is very common to use the Observer pattern within GUI frameworks, such as Java's Swing framework. Why do you need to be concerned about how long it takes to process a notification by a GUI element? Should this be the application designer's concern or be dealt with by the framework itself? How can the concern be eliminated?

8. Argue both for and against including the functions to handle the Composite's children in the Component interface. What are the implications for implementation of the pattern and on the intention of the pattern?

Only ONE submission per group is expected. Please submit the answers in a single pdf document via Moodle. Please ensure that the names and roll numbers of the team members are part of the submission sheet.