# UNIT – 2  QUESTIONS (Due: March 14th, 5:00 pm)

An *appropriate* use of diagrams is highly recommended when answering these questions. Create a document named **Unit-Questions-N.pdf**, where **N** is your team's number, containing the questions and your team's answers.

1. Consider the interface **java.util.Collection**, which is implemented by the java collection classes. One of the methods defined in the interface is **iterator()**. Referring to the intention for the pattern and its structure why is **iterator()** a Factory Method?

2. Defend or refute the statement "A factory method is an example of a very simple template method."

3. What does it mean to say a memento object is opaque to its client?  What implications does this have on the client's use of a memento?

4. What issues must be considered if State objects are to be shared by multiple instances of the same Context class (hint: consider mutability)?

5. For Flyweight to be effective, only the Flyweight factory can be allowed to construct new Flyweight objects. How can this be accomplished in Java (hint: consider making the factory a static method in the Flyweight class itself!).

6. Defend or refute the statement "Flyweight objects must be immutable (i.e., unchangeable) once they are created."

7. Discuss the issues involved in deciding whether cloning a prototype is done with shallow or deep copy

8. Explain how the structure of the Interpreter pattern makes use of another pattern you have studied. Show the associations between the participants in the two patterns.

9. Decorator claims to provide a pay-as-you-go approach to adding features.  Do you agree that you can not get this same effect using class inheritance?  Why do you agree or disagree?

10.    Argue against the statement "Coordinating behavior should be defined in the objects being coordinated". How does the Mediator pattern fit into this argument?