**Group 6 1444 – 1451 – 1458**

Answer the following questions:

1. Why interface is used instead of inheritance? Justify your answer.

Interface is used to create an object lets subclasses decide which class to instantiate.

This lets a class differ instantiation to subclasses. To create an object without exposing the creational logic to client.

1. Which part of implementation explains the fact that, the object is created without exposing creation logic to client?

The client asks the ShapeFactory class by creating its object instance to create an object that client asks and then ShapeFactory class will call the interface to create an object of its subclass.

1. What does the client do?

The client creates the object of ShapeFactory class and calls its methods to create object.

1. Which is the common interface used to refer objects.

The interface shape is used to refer objects.

1. With respect to the example provided, which is the super-class and sub-classes?

The super class will be the interface Shape and

The sub class will be Circle, Square and rectangle.

1. Based on information passed, the object is created and passed. Explain this statement with respect to the example provided.

The client class will pass the information to the ShapeFactory class to create an object and return it back to client class to perform further operations on that object.

1. Assign the following roles to the components of the example provided:
   1. Product - Shape
   2. ConcreteProduct – Circle, Shape and Rectangle
   3. Creator - FactoryPatternDemo
   4. ConcreteCreator – Shapefactory
2. What is the advantage of putting the object creation code into another class called factory?

To create an object without exposing the creational logic to client i.e FactoryPatternShape.

Underlying implementation can be changed without any impact on caller api.

1. What if the factory is defined as “static”?

If factory is defined as static it will create only one instance of object at a time and that will be accessible to all the sub classes.

1. Can you inherit a constructor? How this issue is related to factory pattern?

It is not possible to inherit constructor as subclass has different class names but we can call the parent constructor by using the keyword “super” .

This issue is handled in factory pattern as factory provides the access to the objects which clients require.

**SUMMARY:**

Factory pattern is a creational pattern based on encapsulation which provides to create object.

It also provides to create object without the knowledge of the Client class.

It refers to the newly created object through a common interface.

Factory pattern allows us to create objects without specifying the exact class of object.

Its not necessary that the user should know the creational part subclasses.