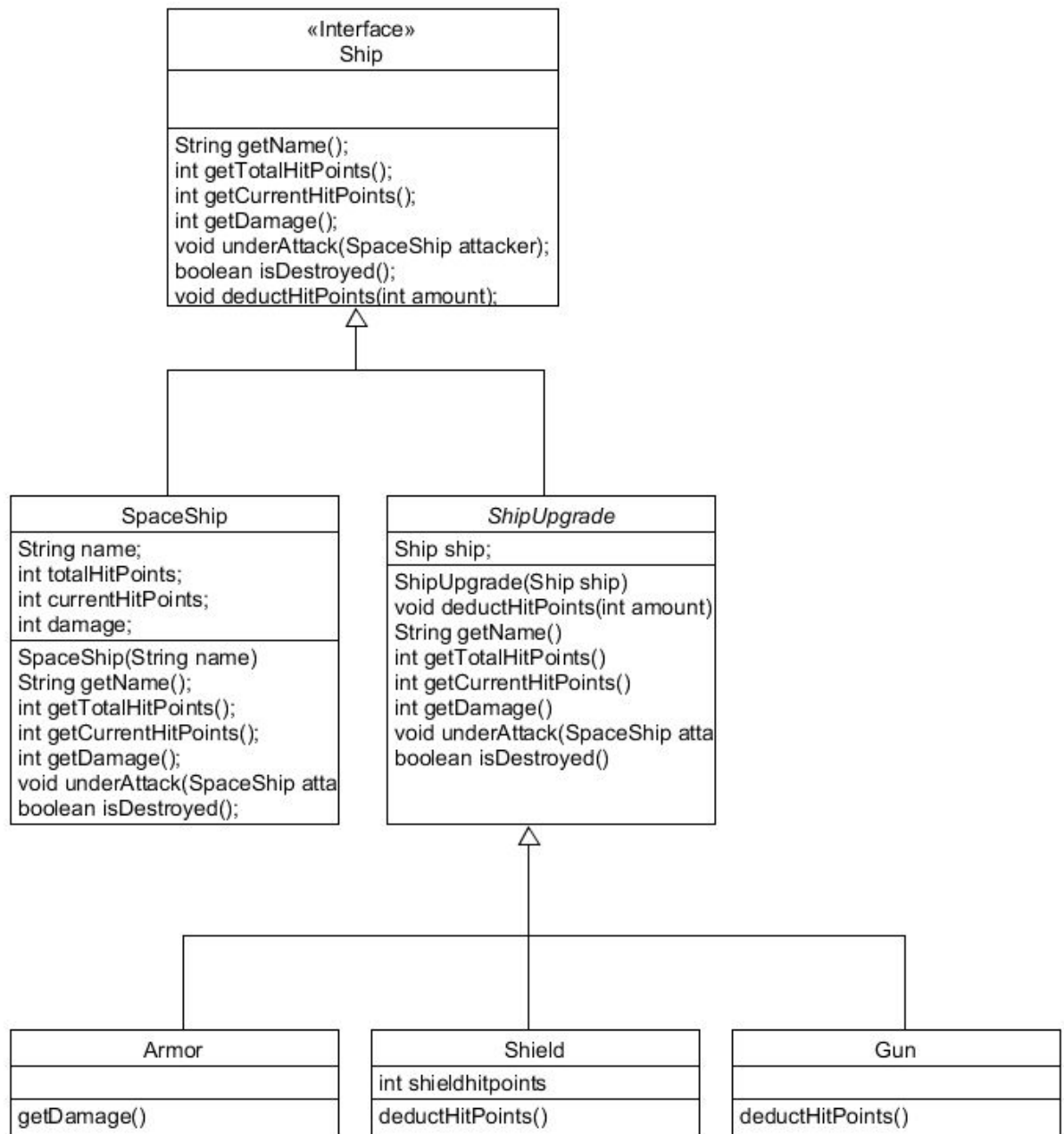# SWEN 262 Refactoring Exercise

INSTRUCTIONS: Analyze the Java code provided to you and read the instructions below. Submit a ZIP archive containing a PDF version of this document that includes the required artifacts in the space after each question and your refactored source code.

1. Fill out the **GoF Pattern Card** below to describe the classes that you would use to refactor the Java class into the required design pattern. You should write *at least 2-3 sentences per class* to describe its role in the pattern in the context of the application.

Gof Pattern Name: Decorator		
Class	**GoF Participant Name**	**Participant's activity within the pattern in the context of the application.**
Ship	Component	This is an interface that is implemented by spaceship and shipUpgrade and has all the methods that a client needs to call.
Spaceship	Concrete Component	This is a concrete implementation of Ship that contains all the standard normal results of the method calls.
ShipUpgrade	Decorator	This is an abstract class that the concrete decorators extend and has an basic decorator implementation of all the methods from Ship.
Shields,Guns,Armor	Concrete Decorator	These Concrete Decorators override and change the methods from Decorator to change their function.
Deviations from the standard pattern: I made ship an interface and ShipUpgrade and abstract class because there should not be any instantiations of those classes.		

2. Provide a UML diagram of the classes and interfaces described in your GoF pattern card. While you may submit a photo or scan of a hand drawn diagram, it must be LEGIBLE.

```
                        «Interface»
                          Ship

                    String getName();
                    int getTotalHitPoints();
                    int getCurrentHitPoints();
                    int getDamage();
                    void underAttack(SpaceShip attacker);
                    boolean isDestroyed();
                    void deductHitPoints(int amount);
```

```
            SpaceShip                              ShipUpgrade

   String name;                           Ship ship;
   int totalHitPoints;                    ShipUpgrade(Ship ship)
   int currentHitPoints;                  void deductHitPoints(int amount)
   int damage;                            String getName()
   SpaceShip(String name)                 int getTotalHitPoints()
   String getName();                      int getCurrentHitPoints()
   int getTotalHitPoints();               int getDamage()
   int getCurrentHitPoints();             void underAttack(SpaceShip atta
   int getDamage();                       boolean isDestroyed()
   void underAttack(SpaceShip atta
   boolean isDestroyed();
```

```
        Armor                    Shield                    Gun

                         int shieldhitpoints
   getDamage()           deductHitPoints()         deductHitPoints()
```

3. Refactor the source code according to your design above. The code must pass any tests that have been provided. Include your refactored code as part of your submission.