Question 1: Basic Shape Drawing Using Interface

1. Define an interface named `Drawable` .
2. This interface should declare a method:

```
void draw();
```

3. Create three classes:
   - `Circle`
   - `Rectangle`
   - `Triangle`
4. Each of these classes should implement the `Drawable` interface.
5. Implement the `draw()` method in each class. The method should print a message indicating which shape is being drawn:
   - For example:
     - `"Drawing a Circle"`
     - `"Drawing a Rectangle"`
     - `"Drawing a Triangle"`

6. **Test**:
   - Create individual objects of `Circle` , `Rectangle` , and `Triangle` .
   - Declare a reference of type `Drawable` and assign it to each object one by one.
   - Call the `draw()` method through the interface reference to demonstrate polymorphism.

Question 2: Vehicle Operations Using Interface

1. Define an interface named `Vehicle` .
2. The interface should declare the following methods:

```
void startEngine();
void stopEngine();
void accelerate(int amount);
int getCurrentSpeed();
```

3. Create two classes:
   - `Car`
   - `Motorcycle`
4. Both classes should implement the `Vehicle` interface.
5. Each class should maintain an internal variable, such as `int speed` , to represent the current speed.
6. Implement the interface methods in both classes:
   - `startEngine()` : Print a message indicating the engine has started.
   - `stopEngine()` : Print a message indicating the engine has stopped and reset speed to 0.
   - `accelerate(int amount)` : Increase the speed variable by the given amount.
   - `getCurrentSpeed()` : Return the current speed.

7. **Test**:
   - Declare a reference of type `Vehicle` .
   - Assign it to a `Car` object and call all methods.
   - Assign it to a `Motorcycle` object and call all methods.

Question 3: Data Management with Multiple Interfaces

1. Define an interface named `Storable` with the following method:

   ```java
   boolean save();
   ```

2. Define another interface named `Loggable` with the following method:

   ```java
   void logMessage(String message);
   ```

3. Create a class named `UserProfile` with the following properties:
   - `String username`
   - `int userID`

4. The `UserProfile` class should implement both the `Storable` and `Loggable` interfaces.

5. Implement the methods in the `UserProfile` class:
   - `save()` : Print a message like `"Saving user [username]..."` and return `true` .
   - `logMessage(String message)` : Print a message like `"[username]: [message]"` .

6. **Test**:
   - Create an object of type `UserProfile` .
   - Call the `save()` method.
   - Call the `logMessage()` method.
   - Assign the object to a `Storable` reference and invoke `save()` .
   - Assign the same object to a `Loggable` reference and invoke `logMessage()` .