1. A list of elements of a given type can be represented using an array of that type.
2. A list of elements of disparate types can be implemented without polymorphism two ways, at least. Such programs share a program structure like _____ (~4–8 lines of pseudo-code).

   ```
   if( type == …

   else if( type == …

   else if( type == …
   ```

3. **Stumbling block**

   This structure constitutes a maintenance nightmare: _____ (~20–60 words).

   Adding new types that are allowed in the list requires an extension of the pattern code in every spot in the class that has the pattern. It is also a violation of NTTSTT. The parallel array implementation also stores information rather ineffectively

4. Polymorphism provides programmers with a workaround for the stumbling block: implement a list of elements of disparate types by _____ (~20–40 words).

   having a general Object class and subclasses of types. An Object list can be created such that a disparate type can be held in the list because a subclass is-A Object.

5. Java features support applying a method to every element when all of the types have an implementation of that method.

   For example, there may exist a Dog class that has a speak method wherein the Dog barks. There may also be a subclass Terrier that is-A Dog. (~20–40 words). The implementation can be defined in the class that represents the type, as in Dog. Alternatively, the implementation can be inherited, the way class Terrier inherits method bark from class Dog. These examples show that it is immaterial *how* the method becomes part of the class.