```
events.addElement(new TimedFinalEvent("500free.txt", 6));
events.addElement(new PrelimEvent("100free.txt", 6));
```

Clearly, this is an instance where an EventFactory may be needed to decide which kind of event to generate. This revisits the simple factory we began the discussion with.

When to Use a Factory Method

You should consider using a Factory method under the following circumstances:

- A class can't anticipate which kind of class of objects that it must create.
- A class uses its subclasses to specify which objects it creates.
- You want to localize the knowledge of which class gets created.

There are several variations on the Factory pattern.

1. The base class is abstract, and the pattern must return a complete working class.
2. The base class contains default methods and is subclassed only when the default methods are insufficient.
3. Parameters are passed to the factory telling it which of several class types to return. In this case, the classes may share the same method names, but each may do something quite different.

Thought Question

1. Seeding in track is carried out from inside to outside lanes. What classes would you need to develop to carry out track-like seeding?

Programs on the CD-ROM

Program	Description
`\Factory\Factory Method \ShowSeeding.java`	Illustrates the Factory Method pattern.