

## Generic Exception Restriction

A generic class cannot extend **Throwable**. This means that you cannot create generic exception classes.

## Continuing Your Study of Generics

As mentioned at the start, this module gives you sufficient knowledge to use generics effectively in your own programs. However, there are many side issues and special cases that are not covered here. Readers especially interested in generics will want to learn about how generics affect class hierarchies, run-time type comparisons, and overriding, for example.

Discussions of these and other topics are found in my book *Java: The Complete Reference, J2SE 5 Edition* (McGraw-Hill/Osborne, 2005).



## Module 13 Mastery Check

1. Generics are an important addition to Java because they enable the creation of code that is
  - A. Type-safe
  - B. Reusable
  - C. Reliable
  - D. All of the above
2. Can a primitive type be used as a type argument?
3. Show how to declare a class called **FlightSched** that takes two generic parameters.
4. Beginning with your answer to question 3, change **FlightSched**'s second type parameter so that it must extend **Thread**.
5. Now, change **FlightSched** so that its second type parameter must be a subclass of its first type parameter.
6. As it relates to generics, what is the ? and what does it do?
7. Can the wildcard argument be bounded?
8. A generic method called **MyGen()** has one type parameter. Furthermore, **MyGen()** has one parameter whose type is that of the type parameter. It also returns an object of that type parameter. Show how to declare **MyGen()**.

9. Given this generic interface

```
interface IGenIF<T, V extends T> { // ...
```

show the declaration of a class called **MyClass** that implements **IGenIF**.

10. Given a generic class called **Counter<T>**, show how to create an object of its raw type.
11. Do type parameters exist at run time?
12. Convert your solution to question 10 of the Mastery Check for Module 9 so that it is generic. In the process, create a stack interface called **IGenStack** that generically defines the operations **push()** and **pop()**.