Java Generics

Parametric Polymorphism

GENERIC METHODS

DR. ERIC CHOU

IEEE SENIOR MEMBER

Generic Methods

A generic type can be defined for a static method

- You can define generic interfaces (e.g., the **Comparable** interface in Figure B(b) and classes (e.g., the **GenericStack** class in **GenericStack.java**). You can also use generic types to define generic methods. For example,
  - **GenericMethodDemo.java** defines a generic method to print an array of objects.
  - **GenericMethodDemo.<Integer>print(integers)** passes an array of integer objects to invoke the generic print method.
  - **GenericMethodDemo.<String>print(strings)** invokes print with an array of strings.

Declaration of a Generic Method

Generic Instance Method has different way or declaring type variable from Generic Static Method.

- To declare a generic method, you place the generic type **<E>** immediately after the keyword **static** in the method header. For example,

public static <E> **void** print(E[] list)

- To invoke a generic method, prefix the method name with the actual type in angle brackets.

For example,

```
GenericMethodDemo.<Integer>print(integers);
GenericMethodDemo.<String>print(strings);
```

or simply invoke it as follows:

```
print(integers);
print(strings);
```

Generic Method

Demo Program: GenericMethodDemo.java

Go BlueJ!