Generics in Java

Generic Programming = same logic/code for various data types

C language = void*

C++ language = templates

Java (till 1.4) = using Object

Java (5.0 or above) = generics

Generic programming

- generic class
- generic method
- generic interfaces

Generic class

- data structures are idea examples for generic classes
- e.g. stack, queue, linked list, ...

Generic class

Generics advantage: type-safety
type-safety → type check at compile time.

Till Java 1.4

```
class Box {
    private Object obj;
    public void set(Object obj) {
        this.obj = obj;
    }
    public Object get() {
        return this.obj;
    }
}
main():
    Box b1 = new Box();
    b1.set("Hello");
    String r1 = (String) b1.get();
    Box b2 = new Box();
    b2.set("123");
    Integer r2 = (Integer) b2.get();
```

↳ ClassCastException: String → Integer

Since Java 5.0

```
class Box <TYPE> {
    private TYPE obj;
    public void set(TYPE obj) {
        this.obj = obj;
    }
    public TYPE get() {
        return this.obj;
    }
}
main():
    Box<String> b1 = new Box<String>();
    b1.set("Hello");
    String r1 = (String) b1.get();
    Box<String> b2 = new Box<String>();
    b2.set("123");
    Integer r2 = b2.get(); //Compiler error
    String r2 = b2.get(); //Okay
```