

What's new in Java 1.5?



Sid Stamm



New Java Stuff!

- 🍯 Lots of new toys
- 🍯 How does it affect APCS now?
- 🍯 How *will* it affect APCS later?



The New Toys

- 🔸 Generics
- 🔸 Autoboxing/Unboxing
- 🔸 Typesafe Enums
- 🔸 Varargs
- 🔸 Static Import
- 🔸 Annotations (Metadata)



```
List things = createListOfBorkObjects();  
for(Iterator i = things.iterator(); i.hasNext(); ) {  
    Bork item = (Bork)i.next();  
    //do something useful  
}
```

Generics

```
List<Bork> things = createListOfBorkObjects();  
for(Iterator<String> i = things.iterator(); i.hasNext();) {  
    Bork item = i.next();  
    //do something useful  
}
```

Generics!

```
Map<String, Bork> myMap = new HashMap<String, Bork>();
```

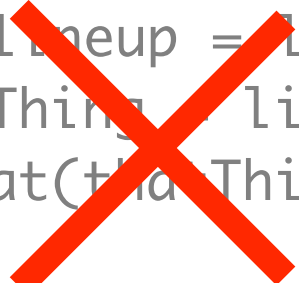
Generics!!

For/in loop

```
for(Iterator lineup = list.iterator(); lineup.hasNext(); ) {  
    Object thatThing = lineup.next();  
    myMonster.eat(thatThing);  
}
```

For/in loop!

```
for(Iterator lineup = list.iterator(); lineup.hasNext(); ) {  
    Object thatThing = lineup.next();  
    myMonster.eat(thatThing);  
}
```



```
for(Object thatThing : list ) {  
    myMonster.eat(thatThing);  
}
```

For/in loop!! (Arrays)

```
int[] nums = { 1, 2, 3, 4, 5, 6 };  
  
for(int n : nums) {  
    System.out.println(n);  
}
```

For/in loop!!! (Becoming Iterable)

```
public class MailBox implements Iterable<MailMessage> {  
  
    /** structure storing the messages */  
    private ArrayList<MailMessage> messages;  
    //...  
    /** Implemented for Iterable. */  
    public Iterator<MailMessage>() {  
        return messages.iterator();  
    }  
    //...  
}
```

```
Integer i = new Integer(4);
```

```
int j = i.intValue();
```

```
Number n = new Float(3.14159);
```

```
Boolean stuff = new Boolean(false);
```

```
// things before the ? must be a boolean (lower case)
```

```
System.out.println( stuff.booleanValue() ? "Yep" : "Nope" );
```

Autoboxing/Unboxing

```
Integer i = 4;
```

```
int j = i;
```

```
Number n = 3.14159f;
```

```
Boolean stuff = false;
```

```
// things before the ? must be a boolean (lower case)
```

```
System.out.println( stuff ? "Yep" : "Nope" );
```

Autoboxing/Unboxing!

Typesafe Enums

```
public enum JettStaff {  
    ADRIAN,  
    ARIJIT,  
    BETH,  
    ERIC,  
    KATIE,  
    KATY,  
    RAJA,  
    RICH,  
    SUZANNE  
};
```

```
JettStaff x = JettStaff.SUZANNE;
```

Typesafe Enums!

```
public enum JettStaff {  
    ADRIAN("Adrian German"),  
    ARIJIT("Arijit Sengupta"),  
    //...  
    SUZANNE("Suzanne Menzel");  
  
    private String name;  
    public JettStaff(String n) { this.name = n; }  
    public String toString() { return this.name; }  
}
```

```
JettStaff x = JettStaff.SUZANNE;  
System.out.println(x);
```

Typesafe Enums!!

```
public enum JettStaff {
    ADRIAN("Adrian German") {
        public String toString() {
            return name + " (dgerman@indiana.edu)";
        }
    },
    //...

    private String name;
    public JettStaff(String n) { this.name = n; }
}
```

```
JettStaff x = JettStaff.SUZANNE;
System.out.println(x);
```



```
public class VarArgs {  
    public static void main(String[] args) {  
        String[] newArgs = {"a", "b", "c"};  
        vaMethod(newArgs);  
    }  
  
    public void vaMethod(String[] args) {  
        System.out.println("You gave me " + args.length  
            + " args! Yay.");  
    }  
}
```

Varargs

```
public class VarArgs {  
    public static void main(String[] args) {  
  
        vaMethod("a", "b", "c");  
    }  
  
    public void vaMethod(String... args) {  
        System.out.println("You gave me " + args.length  
            + " args! Yay.");  
    }  
}
```

Varargs!


```
public class VarArgs {  
    public static void main(String[] args) {  
  
        vaMethod("a", "b", "c");  
    }  
  
    public void vaMethod(String... args) {  
        for(String s : args)  
            System.out.println(s);  
    }  
}
```

Varargs!! (Plus for/in loop)

Static Import

```
import java.awt.*;
```

```
public interface BouncingBallConstants {  
    public static final Color BACK_COLOR = Color.WHITE;  
    public static final Color BALL_COLOR = Color.BLUE;  
    public static final int BALL_SIZE = 50;  
}
```



```
import static Ball;
```

```
public class MyClass extends Object {  
    @Override  
    public String toString() {  
        return "My overridden method!";  
    }  
}
```

Annotations (Metadata)

```
public @interface Conference {
    String what();
    String when();
    String location();
}

@Conference(what="JETT",
            when="November 2004",
            location="IUB")
public class MyMagicJettClass {
    //...
}
```

Annotations (Metadata)!



**How does it affect
APCS now?**





**How *will* it affect
APCS later?**



References

- ❁ ***J2SE(tm) 5.0 New Features.*** Sun Microsystems.
<http://java.sun.com/j2se/1.5.0/docs/relnotes/features.html>
- ❁ ***New Language Features for Ease of Development in the Java 2 Platform, Standard Edition 1.5: A Conversation with Joshua Bloch.*** Heiss, Janice J.
May 8, 2003. Sun Microsystems.
http://java.sun.com/features/2003/05/bloch_qa.html
- ❁ ***Java 1.5: Where does it fit into AP Computer Science?***
Weiss, Mark. Florida International University. AP Central & Collegeboard.com.
<http://apcentral.collegeboard.com/members/article/1,3046,151-165-0-36930,00.html>
- ❁ ***Java 1.5 Tiger: A Developer's Notebook.*** McLaughlin, Brett & Flanagan, David.
O'Reilly Media, Inc, 2004 ISBN 0-596-00738-8.