Java™ Intermediate
Version 1.2

Copyright (c) 1997-2004 N Afshartous - All rights reserved. This product and related documentation are protected by copyright and distributed under licenses restricting its use, copying, distribution, decompilation. No part of this product or related documentation may be reproduced in any form by any means without prior written authorization.

Trademarks: Java, Java Development Kit are trademarks of Sun Microsystems. Mosaic, NCSA Mosaic are proprietary trademarks of the University of Illinois.

Disclaimer: Although great care has been taken to ensure the accuracy and quality of these materials, all material is provided without any warranty whatsoever, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose.
About the author:  Niki Afshartous has been teaching corporate technical training classes for five years. Previously he was at the MIT Lincoln Laboratory as a systems programmer. He holds a Ph.D. degree in computer science from New York University where his research focus was on concurrent programming languages.

Dedication:  For Jenny

Comments, questions? Don’t hesitate to contact the author.
Email: afshar@cs.nyu.edu
WWW: www.cs.nyu.edu/afshar
Chapter 0: Review

After this chapter you will be able to:

- Remember what you learned in Java™ Intro
Questions

• What is the meaning of a type?
• If $x$ is an integer variable then what can we say about $x$?
Answer

• A variable’s type implies **both:**
  – the legal operations that may be applied to the variable
  – the range of values that may be stored in the variable

• If $x$ is an integer then:
  – we can apply arithmetic operations to $x$
  – we can store integer values in $x$

Types have similar implications for object variables.
Question

• Can x store a value or an address?

```c
int x;
```
Answer

• Variables of primitive type store values
  – primitive types: int, float, char, double, boolean, short
Question

• Can variable s store a value or an address?

```java
String s;
```
Answer

• Variables of object type store addresses
  – object types are created by class definitions

• Java has predefined classes such as:
  – String, Integer, Float, Vector
Question

• Does this statement create an object?

String s;
Answer

• No

```java
String s;
```

• Variable `s` can store the address of an object

```java
s
```
Question

• Does this statement create an object?

```java
String[] sArray;
```
Answer

- No

```
String [] sArray;
```

- Variable `sArray` can store the address of an array object
Question

• How many objects does this create?

String [] b = {"hi", "there";}
Answer

• Three: an array object and two String objects

```java
String [] b = {"hi", "there"};
```

![Diagram showing an array object `b` containing two String objects `hi` and `there`.]
Question

• Are objects stored in the stack or the heap?
Answer

• In Java, all objects are stored in the heap

• In C++, objects can be stored in either the stack or the heap
Question

• What does the Garbage Collector (GC) do?
Answer

• The GC frees up the storage occupied by objects no longer accessible by the program
GC

• The GC will not reclaim as long as an object is accessible

```java
public static void main (String [] args) {
    int [] x;
    {
        int [] y = {1,2,3};
        x = y;
    }
}
```

• Question: Is it legal for the GC to reclaim the array object at this point?
Answer

• No, since the object is still accessible via $x$

```java
public static void main (String [] args) {
    int [] x;
    {
        int [] y = {1,2,3};
        x = y;
    }
}
```

Stack

Heap

$1,2,3$
OOAD Step 1

• List the entities in the application domain

- Console
- Plane
- Passenger cabin
- Tires
- Missiles
- Wing
- Jetfighter
- Engine
- Airliner
- Exhaust
- Cockpit
- Altimeter
- Pilot seat
- Passenger seat
- Landing gear
- Seat
- Pilot seat

• What relationships do you see?
• Identify the inheritance (Is-A) relationships
OOAD Step 3

- Identify the containment (Has-A) relationships

Diagram:
- Plane
  - Cockpit
    - Console
  - Engine
    - Exhaust
  - Wing
  - Landing gear
    - Tires
  - Pilot seat
  - Altimeter