

*Details of Java Object Passing*

## CS 351

# Design of Large Programs

Instructor: **Joel Castellanos**

**e-mail:** [joel@unm.edu](mailto:joel@unm.edu)

**Web:** <http://cs.unm.edu/~joel/>

**Office:** Farris Engineering  
Center (FEC) room 321

Lab Instructor: **David Godinez**

**e-mail:** [dgodinez@cs.unm.edu](mailto:dgodinez@cs.unm.edu)



2/2/2009

## Allocating Space for Arrays of Objects

```
public static void main(String[] args)
{ //Allocates space for three 32-bit integers.
  int[] x = new int[3];

  //Allocates space for three references.
  Point[] p = new Point[3];

  for (int i = 0; i<3; i++)
  { x[i] = i;

    //Create each of the pointer objects.
    p[i] = new Point(i,i*2);
  }

  for (int i = 0; i<3; i++)
  { System.out.println(x[i] + ", " +
    p[i].x + ", " + p[i].y);
  }
}
```

2

## #1: What is the Output?

```
public class HelloWorld
{ public static void foo(Point tmp)
  { tmp.x = 5;
    tmp.y = 25;
  }

  public static void main(String[] args)
  { Point a = new Point(2,4);
    Point b = new Point(3,9);
    Point c = a;

    foo(a);

    System.out.println(a.x + ", " + a.y);
    System.out.println(b.x + ", " + b.y);
    System.out.println(c.x + ", " + c.y);
  }
}
```

3

## #1: Answer

```
public class HelloWorld
{ public static void foo(Point tmp)
  { tmp.x = 5;
    tmp.y = 25;
  }

  public static void main(String[] args)
  { Point a = new Point(2,4);
    Point b = new Point(3,9);
    Point c = a;

    foo(a);

    System.out.println(a.x + ", " + a.y);
    System.out.println(b.x + ", " + b.y);
    System.out.println(c.x + ", " + c.y);
  }
}
```

**2** a, c, tmp → Point(5,25)  
b → Point(3,9)

**1** a, c → Point(2,4)  
b → Point(3,9)

**3** 5, 25  
3, 9  
5, 25

4

## #2 What is the Output?

```
public class HelloWorld
{ public static void foo(Point tmp)
  { tmp = new Point(5,25);
  }

  public static void main(String[] args)
  { Point a = new Point(2,4);
    Point b = new Point(3,9);
    Point c = a;

    foo(a);

    System.out.println(a.x + ", " + a.y);
    System.out.println(b.x + ", " + b.y);
    System.out.println(c.x + ", " + c.y);

  }
}
```

5

## #2 Answer

```
public class HelloWorld
{ public static void foo(Point tmp)
  { tmp = new Point(5,25);
  }

  public static void main(String[] args)
  { Point a = new Point(2,4);
    Point b = new Point(3,9);
    Point c = a;

    foo(a);

    System.out.println(a.x + ", " + a.y);
    System.out.println(b.x + ", " + b.y);
    System.out.println(c.x + ", " + c.y);

  }
}
```

**2** a, c → Point(2,4)  
b → Point(3,9)  
tmp → Point(5,25)

**1** a, c → Point(2,4)  
b → Point(3,9)

**3** a, c → Point(2,4)  
b → Point(3,9)  
<garbage collect> Point(5,25)

6

### #3 What is the Output?

```
public class HelloWorld
{ public static void foo(Point[] tmp)
  { tmp[1] = new Point(5,25);
  }

  public static void main(String[] args)
  { Point[] p = new Point[2];
    p[0] = new Point(2,4);
    p[1] = new Point(3,9);
    Point c = p[1];

    foo(p);

    System.out.println(p[0].x + ", " + p[0].y);
    System.out.println(p[1].x + ", " + p[1].y);
    System.out.println(c.x + ", " + c.y);
  }
}
```

7

### #3 What is the Output?

```
public class HelloWorld
{ public static void
      foo(Point[] tmp)
  { tmp[1] = new Point(5,25);
  }
}
```

**2** p, tmp → { p[0], p[1] }  
p[0] → Point(2,4)  
c → Point(3,9)  
p[1] → Point(5,25)

```
public static void main(String[] args)
```

```
{ Point[] p = new Point[2];
  p[0] = new Point(2,4);
  p[1] = new Point(3,9);
  Point c = p[1];
```

**1** p → { p[0], p[1] }  
p[0] → Point(2,4)  
p[1], c → Point(3,9)

```
foo(p);
```

**3** p → { p[0], p[1] }  
p[0] → Point(2,4)  
c → Point(3,9)  
p[1] → Point(5,25)

```
}
```

8

## Quiz 1-5: What is the Output?

```
public static void foo(Point[] tmp)
{ tmp[0] = new Point(4,64);
  tmp = new Point[7];
}
public static void main(String[] args)
{ Point[] p = new Point[2];
  p[0] = new Point(2,8);
  p[1] = new Point(3,27);
  Point c = p[0];
  foo(p);
  System.out.println(
    p[0].x + ", " + p[1].x + ", " + c.x);
}
```

- a) 4, 3, 2
- b) 4, 3, 3
- c) 4, 2, 2
- d) 4, 3, 4
- e) 0, 0, 0

9

## SerialVersionUID Warning

Serializable class does not declare a static final field serialVersionUID of type long.

```
private static final long serialVersionUID = 1L;
```

**Object serialization** provides a program the ability to read or write *entire objects* to and from a network or file-system without writing code to convert to/from raw bytes or parsing clumsy text data.

```
File myFile = new File("myDataFile");
```

```
// Setup a stream to a physical file.
```

```
FileOutputStream outputStream = new FileOutputStream(myFile);
```

```
// Attach a stream capable of writing objects.
```

```
ObjectOutputStream objStream = new
  ObjectOutputStream(outputStream);
```

10

```
objStream.writeObject(teamObj[i]);
```

## #1: Comparing Strings: What Output?

```
String a = new String("Dara");
String b = new String("Rowan");
String c = new String("Dara");

if (a == b) System.out.println("Same");
else      System.out.println("Different");

if (a == c) System.out.println("Same");
else      System.out.println("Different");

if (b == c) System.out.println("Same");
else      System.out.println("Different");
```

Output

```
Different
Different
Different
```

11

## #2: Comparing Strings: What Output?

```
String a = new String("Dara");
String b = new String("Rowan");
String c = new String("Dara");

if (a.equals(b)) System.out.println("Same");
else             System.out.println("Different");

if (a.equals(c)) System.out.println("Same");
else             System.out.println("Different");

if (b.equals(c)) System.out.println("Same");
else             System.out.println("Different");
```

Output

```
Different
Same
Different
```

12

### #3: Comparing Strings: What Output?

```
String a = "Dara";  
String b = "Rowan";  
String c = "Dara";  
  
if (a == b) System.out.println("Same");  
else      System.out.println("Different");  
  
if (a == c) System.out.println("Same");  
else      System.out.println("Different");  
  
if (b == c) System.out.println("Same");  
else      System.out.println("Different");
```

Output  
Different  
Same  
Different