Summer 2012 - Final Part 1 – Answer Key

1. Polymorphism

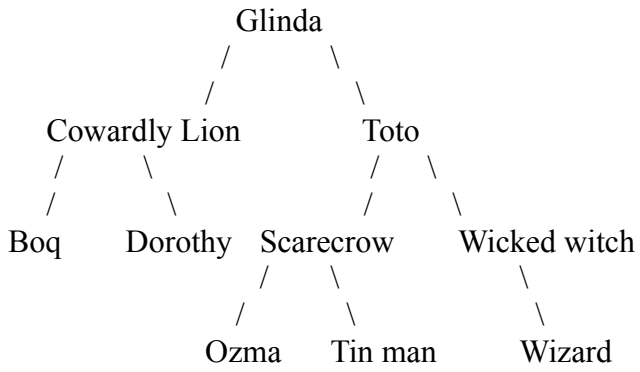
Statement	Output
`var1.a();`	`Teddy 3 / Bear 1 / Bear 3`
`var1.c();`	`Bear 3`
`var2.a();`	`Teddy 3 / Bear 1 / Pooh 3`
`var2.b();`	`Pooh  2/ Bear 3`
`var3.a();`	`Bear 1 / Polar 3`
`var3.b();`	`error`
`var4.a();`	`Teddy 3 / Bear 1 / Pooh 3`
`var5.a();`	`error`
`((Pooh) var5).a();`	`error`
`((Teddy) var1).a();`	`Teddy 3 / Bear 1 / Bear 3`
`((Teddy) var4).a();`	`Teddy 3 / Bear 1 / Pooh 3`
`((Pooh) var3).b();`	`error`
`((Polar) var3).b();`	`Bear 1 / Polar 3 / Polar 2`
`((Teddy) var4).c();`	`Pooh 3`
`((Bear) var5).c();`	`Bear 3`

2. Comparable

```
public class Office implements Comparable<Office> {
	private double width;
	private double length;
	private boolean couch;
	private int windows;

	public Office (int width, int length, boolean couch, int windows) {
		this.width = width;
		this.length = length;
		this.couch = couch;
		this.windows = windows;
	}

	public String toString() {
		String s = "width: " + width + ", length: " + length +
		                                                ", windows: " + windows;
		if (couch) {
			s += ", has a couch";
		}
		return s;
	}

	public boolean isCorner() {
		return length == width && windows == 2;
	}

	public int compareTo (Office other) {
		if (length * width < other.length * other.width) {
			return 1;
		} else if (length * width > other.length * other.width) {
			return -1;
		} else {
			if (windows != other.windows) {
				return other.windows - windows;
			} else {
				if (couch && !other.couch) {
					return -1;
				} else if (!couch && other.couch) {
					return 1;
				} else {
					return 0;
				}
			}
		}
	}
}
```

3. BST

Values to insert:
Glinda, Cowardly lion, Dorothy, Toto, Scarecrow, Tin man, Wicked witch, Wizard, Boq, Ozma



Traversals

Pre-order :
Glinda, Cowardly lion, Boq, Dorothy, Toto, Scarecrow, Ozma, Tin man, Wicked witch, Wizard

In-order :
Boq, Cowardly lion, Dorothy, Glinda, Ozma, Scarecrow, Tin man, Toto, Wiked witch, Wizard

Post-order :
Boq, Dorothy, Cowardly lion, Ozma, Tin man, Scarecrow, Wizard, Wicked witch, Toto, Glinda

4. Binary Trees

```
public int sumLeaves() {
      return sumLeaves(overallRoot);
}

private int sumLeaves(IntTreeNode node) {
      if (node != null) {
            if (node.left == null && node.right == null) {
                  return node.data;
            } else {
                  return sumLeaves(node.left) + sumLeaves(node.right);
            }
      }
      return 0;
}
```