1999 Foundations of Programming

p10q2
FHK

NOTES ON THE MODEL ANSWER TO PAPER 10 QUESTION 2

The keyword final may be applied to classes, methods and variables and is used most notably for setting up constants. For example:

```
int final heinz = 57;
```

This would make heinz an int constant whose value was 57.

The keyword finally is used in constructs such as:

```
try
   <guarded statements>
catch
   ---
finally
   ---
```

If an exception is thrown in the guarded statements then it may be caught in a catch clause where the statements handle the exception. Whether the exception is thrown or not and whether it is caught or not the statements in the finally clause are executed. These statements are often responsible for essential tidying up.

When all references to an object are lost, the object becomes garbage. At intervals, space occupied by garbage is recovered by the automatic action of the Java garbage collector. Any class may incorporate a special method finalize and when the garbage collector recovers the space of a dead object which contain a finalize method, this method is invoked. The method usually undertakes essential tidying up.

The program creates two instances of a Square object, one called jack and the other called jill. The output is:

```
jack's details: Square 1, edge size 8
jill's details: Square 2, edge size 12
Number of squares: 2
Number of squares: 2
```

If edge is declared static, both jack and jill share the same (static) instance of edge. Accordingly jill's 12 will overwrite jack's 8 and the first two lines of output will be:

```
jack's details: Square 1, edge size 12
jill's details: Square 2, edge size 12
```

The required extra method is:

```
public void finalize()
 { this.total--;
 }
```

The required version of method main() incorporates a throws clause and a Thread.sleep statement:

```
public static void main(String[] args) throws InterruptedException
 { Square jack = new Square(8);
   Square jill = new Square(12);
   System.out.println("jack's details: " + jack);
   System.out.println("jill's details: " + jill);
   System.out.println("Number of squares: " + Square.total);
   jack = jill;
   Thread.sleep(1000L);
   System.out.println("Number of squares: " + Square.total);
 }
}
```