Solution notes

Foundations of Programming (Java) 2005 – Paper 11 Question 2 (FHK)

THE INNER CLASSES QUESTION

Context: Much of this question stems from MODULE10p of the
Foundations of Programming Course.

Part (a). Bookwork. The three kinds of inner class are
Member Classes, Local Classes and Anonymous Classes. Class
ClassIn in the program is a Member Class. [4 marks]

Part (b). The condition this.b provokes the compiler error.
There is no data field b in class ClassIn so this.b is
invalid. The data field b referred to is in the instantiation
of the outer class Trial. Accordingly, it should be accessed
either by Trial.this.b or simply by b alone. Either version
can be used as the condition. Note that the local boolean
variable b in the constructor ClassIn() is a red herring.
[4 marks]

Part (c). With either version of this fix the program will
compile and run. At run-time, method main() sets up an object
of type Trial at which stage the boolean instance variable b
is set set to true and the constructor Trial() sets up an
instance of the inner class ClassIn.

At this stage the constructor ClassIn() declares and initialises
a boolean local variable (also called b) to false but this variable
is never used. The method main() not only sets up the object but
also invokes the method test(). It is here that Trial.this.b is
tested and is still true. Accordingly Yes is printed.

After Yes is printed the constructor Trial() sets this.b false
but this is too late to have an effect. This assignment is
another red herring. [6 marks]

Part (d). The program can readily be rearranged so that class
ClassIn is a local class within the constructor Trial() as in:

```
public class Trial
 { private boolean b = true;

   private Trial()
    { class ClassIn
       { ClassIn()
```

```
            { boolean b = false;
            }

           private void test()
            { if (Trial.this.b)
                 System.out.println("Yes");
              else
                 System.out.println("No");
            }
        }

      (new ClassIn()).test();
      this.b = false;
    }

   public static void main(String args [])
    { new Trial();
    }
}
```

The only point of note is that visibility modifiers which are inappropriate for local variables are equally inappropriate for local classes. Accordingly, the modifier private which previously heralded class ClassIn has been removed. [6 marks]