4 Object-Oriented Programming (acr31)

(a) Describe the differences between primitive types and objects in Java. Consider:

(i) the values they contain [1 mark]

(ii) where they are stored in memory [1 mark]

(iii) how they interact with Java references [1 mark]

(b) What are auto-boxing and auto-unboxing? Give an example of how they might cause an exception to be thrown. [4 marks]

(c) Consider the following code in which any arbitrary Java type (primitive or object) could be substituted for \( T \).

\[
\begin{align*}
\text{void } f&(\text{T } t) \{ /* ... */ \} \\
\text{T } &t1 = /* ... */ \\
&f(t1);
\end{align*}
\]

For which substitutions of \( T \) can we guarantee that the value in \( t1 \) is unchanged after the invocation of \( f(t1) \)? Justify your answer. [3 marks]

(d) Explain how Java’s implementation of generics precludes substituting \( T \) with a primitive type. [2 marks]

(e) You are asked to redesign the standard library to incorporate an immutable list. Explain the relative merits of:

(i) MutableList being a subtype of ImmutableList [2 marks]

(ii) ImmutableList being a subtype of MutableList [2 marks]

(iii) ImmutableList and MutableList having no common supertype [2 marks]

(iv) ImmutableList and MutableList both subtyping CommonList [2 marks]