

4 Object-Oriented Programming (acr31)

A program which decrypts files under the Swap Encryption Scheme by swapping pairs of characters is given below. Some code has been omitted and you do not need to understand the operation of the algorithm.

```
1  class Swapper extends Reader {
2      private final PushbackReader pushBack;
3      Swapper(PushbackReader p) { pushBack = p; }
4
5      @Override
6      public int read(char[] cbuf, int off, int len) {
7          int r = wrap.read(cbuf, off, len);
8          if (r % 2 == 1) { pushBack.unread(cbuf, off + --r, 1); }
9          for (int i = 0; i < r; i += 2) { swap(cbuf, i, i + 1); }
10         return r;
11     }
12 }
13
14 class Decryptor {
15     static List<String> read(String fileName) {
16         try (BufferedReader r = new BufferedReader(new Swapper(
17             new PushbackReader(new FileReader(fileName)))) {
18             return readLines(r);
19         }
20     }
21 }
```

- (a) The four principles of object-oriented programming are encapsulation, abstraction, inheritance of code and polymorphism. Explain how the program above makes use of each of them, with reference to specific lines in the code. [2 marks each]
- (b) The program attempts to use `Swapper` as part of the Decorator pattern. What changes would you make to improve the design? [2 marks]
- (c) Explain how this program demonstrates the open-closed principle. [2 marks]
- (d) How would you change this implementation to allow users to specify an arbitrary operation to apply to pairs of characters (rather than just swapping them)? [4 marks]
- (e) Explain why this design does not satisfy the open/closed principle with respect to adding support for decrypting images. What are the implications of this for object-oriented program design? [4 marks]