

**Question 1**

Consider the code snippet below in relation to the SOLID design principles.

```
public class Parser{
    public void parseFile(String format, URI uri, int fileSizeInMB){
        if(format.equals("JSON")){ //original code March 10 2019
            // code to parse file as JSON
        }
        if(format.equals("XML")){
            if(fileSizeInMB >= 500){ // added end of April 2019; to handle large files
                //code to parse large XML file using SAX
            }
            else{ //added in April 10 2019
                //code to parse regular XML file
            }
        }
        if(format.equals("CSV")){ //added in May 10 2019
            // code to parse file as CSV
        }
        if(format.equals("HTML")){ // added just in case; not used currently
            // code to parse file as HTML
        }
        //T0 D0: required in June 2019 - code to parse in OWL file format
    }
}
```

Figure 1

Assuming that there are no errors in the code in Figure 1:

- (a) Explain how the Single Responsibility and the Open/Closed principles are being violated. [2 marks]
- (b) Identify two code smells in the code, one each from the *Bloater* and *Dispensibles* categories, and describe why code refactoring is necessary for each case. [4 marks]
- (c) Discuss how you would refactor the code so that the SOLID principles in (a) are followed and the code smells from (b) are eliminated. [6 marks]
- (d) List the remaining three SOLID principles not mentioned in this question. [3 marks]

TOTAL MARKS: 15 marks