CSCU9Y4 Spring 2018

Programming Language Paradigms
Tutorial 18/2

Reading for this tutorial: Bansal sections 4.1, 4.2.4, 4.3 for the first 7 questions. Chapter 7 for the last two, although this has a lot more detail than you need. Look mainly at the introduction, 7.1 and 7.9.

Alternatively, Chapter 5 of Comparative Programming Languages. Chapter 2 of Comparative Programming Languages will give further context, as will Y4_supplementary_notes_2. Chapters 5 and 6 of Watt, Programming Language Design Concepts, or Chapter 11 and 12 of Sebesta, Concepts of Programming Languages are also useful. Or google "information hiding", "encapsulation", "modularity in programming languages".

1. Although you do not know Pascal, you should be able to work out what is happening in the following outline Pascal program:

```
program Example(input, output);
    type Person = record
        name: String;
        age: Integer;
    end;
    var p1, p2: Person;
    ...
    procedure mkPerson(var p:Person; n:String; a:Integer);
    begin
        p.name := n;
        p.age := a;
    end; {mkPerson}

    procedure haveBirthday(var p:Person);
    begin
        p.age := p.age + 1;
    end; {haveBirthday}

    function getAge(p:Person): Integer;
    begin
        getAge := p.age;
    end; {getAge}

 begin ... main program code goes here ... end.
```

   Write procedure calls so that variable `p1` will refer to a `Person` called `Bill` who is `35`, will then have a birthday and then show how we can find the age of `p1`.

2. Define a Java class `Person` corresponding to the above Pascal code. What would be the Java equivalent to the procedure and function calls of question 1?

3. What is the point of organising programs into blocks and types?

continued over …

4. Two features of object-oriented programming are information hiding and encapsulation. What is meant by these terms? In what way does the above Pascal code not support information hiding? What aspect of encapsulation does it support and what aspect does it not support?

5. How does Ada solve the information hiding and encapsulation problems identified in the Pascal example?

6. What is a side-effect? Why should procedures and functions be self-contained? What is a pure function and why is it especially important that functions are self-contained?

7. Why are methods not usually self-contained? Is this a problem? What is the equivalent of a pure function in an object-oriented language?

8. What is meant by *static typing* and by *strong typing*? What advantages do they give?

9. What is dynamic typing? What are its advantages and disadvantages?

University of Stirling
Computing Science and Mathematics
rgc/ces