NEWCASTLE UNIVERSITY

SEMESTER 2 2016/2017

ADVANCED PROGRAMMING

Time allowed - 3 Hours

Instructions to candidates:

Answer TWO questions from Section A,

and TWO questions from Section B.

The total marks available for this exam are 100.

Marks shown for sub-sections are indicative only.

[Turn Over]

SECTION A

Answer TWO questions in this Section.

Question A1

Consider the following class hierarchy.

```
public abstract class Root {
  public abstract void f();

  public Root() {
    System.out.println(“Root Con. called”);
  }
  public final void fin() {
    System.out.println(“Root.fin() called”);
  }
  public void g() {
    System.out.println(“Root.g() called”);
  }
}

public class Sub extends Root {
  int data;
  public Sub(){
    super();
    data = 1;
    System.out.println(“Sub Con. called”);
  }
  public void f(){
    System.out.println(“Sub.f() called”);
  }
  public void g(){
    System.out.println(“Sub.g() called”);
  }
}
```

a) The class `Root` has been declared as *abstract*. What are abstract classes and what is their use in software development? [3 marks]

b) What is the meaning of the specifier `final` which appears in the heading of the method `fin()`? A novice programmer asks you the question "does Java allow abstract final methods?". Explain, with reasons, your reply. [4 marks]

c) Explain the purpose of the call to `super()` in the first line of the constructor `Sub()`. Is the order of the statements in the `Sub()` constructor important? [4 marks]

d) Explain with reasons the effect of the following statement

```
Root root = new Root();
```

[2 marks]

e) What output is produced by the code below

```
public static void main(String[] args){
  Sub s = new Sub();
  Root r = new Sub();
  s.f();
  s.fin();
  r.g();
  if (s.equals(r)){
    System.out.println(“equal”);
  }
  else { System.out.println(“different”);}
}
```

[6 marks]

f) Explain why the code fragment in e) will compile despite the fact that neither the `Root` nor `Sub` classes contain an explicit equals method. What changes would you need to make to the class `Sub` so that instances were compared by value? [6 marks]

Question A2

a) Describe the meaning of the term *package scope* for a class and for a method. [2 marks]

b) What is an *immutable* class? Give two advantages of immutable classes. [3 marks]

c) Consider the following `Article` and `Journal` classes:

```
public class Article {
    private String text;
    private String author;

    public Article(String text, String author) {
        this.text =text;
        this.author = author;
    }

    public String getText() { return text; }
    public String getAuthor () { return author; }

    public void setText(String text) {
        this.text = text;
    }

    public void setAuthor (String author) {
        this.author = author;
    }
}
```

```
public final class Journal {
    private final Set<Article> edition;
    private final Date pubDate;

    public Journal(Article[] contents, Date d) {
        edition = new TreeSet<Article>();
        for (int i = 0; i < contents.length; i++)
            edition.add(contents[i]);

        pubDate = d;
    }
    public Set<Article> getArticles() {
        return
        Collections.unmodifiableSortedSet(edition);
    }

    public Date getDate() { return pubDate; }

}
```

i) Explain the meaning of the `final` qualifier in the class declaration and in the declaration of the member fields of the `Journal` class. [3 marks]

ii) Explain why instances of both `Article` and `Journal` are mutable. [3 marks]

iii) Re-write the `Journal` class to make it immutable (without modifying the `Article` class). [10 marks]

iv) Briefly describe the changes necessary to make `Article` immutable. If these changes were made, would it still be necessary to re-write the `Journal` class as you did in iii) above? Briefly explain your answer. [4 marks]

Question A3

a) Describe and illustrate with a diagram the structure of a *hash set* and a *tree set*. [5 marks]

b) It is possible to use hashing or trees to implement a `Map` interface. Preferably using the Big ‘O’ notation, compare and contrast the performance of the two data structures for the following operations:

   i) get or set a value at a given key in the map,

   ii) insert or remove a key at an arbitrary position in the map.

   Explain the conditions under which worst-case performance will arise for the hashing and tree implementations. [10 marks]

c) A `List` may be implemented using either a resizable array or a linked list of elements. For each of the following scenarios, explain carefully which of these two implementations would be the most efficient to use and why.

   i) A queue of people at a cinema ticket office.

   ii) The names in a mobile phone address book.

   [6 marks]

d) A `Set` would be a more suitable data structure than a `List` for the names in a mobile phone address book (the second scenario from part c) above). State the reason for this and also which of the Java `Set` implementations you would use if you were to implement this and why. [4 marks]

SECTION B

Answer TWO questions in this Section.

Question B1

a) Outline the steps that a Java programmer should take in order to define and start a new thread of concurrent activity. [5 marks]

b) Write a program that creates three threads: one to print 10 times letter X, second to print 5 times letter Y, and the third one to print out integers from 2 to 8. After creating the three threads, the main program should terminate. [15 marks]

c) In what ways is it possible for two threads to access the same object, and what steps should be taken to avoid interference? [5 marks]

Question B2

a) What are the three key features of a *monitor* and how are they used to support concurrent programming? [6 marks]

b) University students are eligible for grants from the university hardship fund in case of financial difficulties. Occasionally, the fund receives donations from generous benefactors. Write a Java monitor to control the fund's budget, remembering that the grants should only be given to students if there is enough money in the fund. [12 marks]

c) Provide an implementation of a `Student` class (or a `Benefactor` class) describing the behaviour of student threads (or benefactor threads), which access the fund from part b). [7 marks]

Question B3

a) What are the distinguishing characteristics of an event driven program as opposed to a conventional program?

[4 marks]

b) Describe briefly the basic steps that a Java programmer needs to take in order to set up event handlers and deal with events. [6 marks]

c) Design a simple event-driven program with a graphical user interface that allows the user to enter the width and the length of a rectangle into text boxes, and then calculates and displays the area of the rectangle. Describe how you would construct such a program, by identifying the basic Java Swing components you would use, and specifying how events are handled. Provide outline code for the relevant parts of the application. [15 marks]