

A20487

THIS IS NOT AN OPEN BOOK EXAMINATION
– CANDIDATES MAY NOT CONSULT ANY
REFERENCE MATERIAL DURING THE
SITTING

No calculator permitted in this examination

THE UNIVERSITY OF BIRMINGHAM

Degree of BSc with Honours
Artificial Intelligence and Computer Science
Computer Science. First Examination
Computer Science/Software Engineering with Business Studies. First Examination
Computer Science/Software Engineering with Business Management First Examination

Degree of BEng/MEng with Honours
Computer Science/Software Engineering. First Examination

Joint Degree of BA with Honours
Computer Studies and English. First Examination

Joint Degree of BEng/MEng with Honours
Computer Science and Civil Engineering. First Examination

Joint Degree of BSc/MSci with Honours
Mathematics and Computer Science. First Examination
Pure Mathematics and Computer Science. First Examination

Undergraduate Occasional
Mathematics

06 18190

Software Workshop 1

Monday 23rd May 2005 1400 hrs – 1700 hrs

[Answer ALL Questions]

No calculator

Section A.

1. Answer the following questions about the Java programming language.

- (a) Write a line of code to declare an integer variable called *n* and initialize it to 3. [1%]
(b) If *n* has value 3, what is the value of the expression $2+n*2 > 9$? [1%]

- (c) The following for loop has three parts missing from its first line. Write down that first line with the three parts filled in so that when the for loop runs it will print out the numbers 1, 3, 5, 7 and 9.

```
for ( ??? ; ??? ; ??? ) {  
    System.out.println(i);  
}
```

[3%]

- (d) Fill in the gaps in these sentences:

- (i) Objects are usually created using the keyword _____.
(ii) If a class _____ an interface then it must provide definitions for the methods in the interface.

[2%]

No calculator

2. (a) The following method is intended to make an array whose elements are 0.0, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5 and 1.75. It contains *six* errors. For each one, explain what the error is, whether and why the compiler would detect it, and how to correct it.

```
public static void makeArray()
{
    double[] a;
    for (i = 0; i < 8; i = i+1);
    {
        a[i] = i/4;
    }
    return a;
};
```

[10%]

- (b) Suppose IA is an interface with a void method f (); CB is a class that implements IA and has a constructor with no parameters; and CC is a class that extends CB, has a new void method g (), and again has a constructor with no parameters. Of the following lines of Java, which are illegal and why?

```
IA x = new IA();    // line 1
IA y = new CC();    // line 2
y.f();              // line 3
y.g();              // line 4
x.f();              // line 5
CC y1 = y;          // line 6
```

[5%]

No calculator

3. The following static method returns the index of the first occurrence of a value x in an array a of integers.

```
/**
 * Find first position of an element in an array.
 * @param a array within which to find element.
 * @param x element to find.
 * @return first index i with a[i] == x
 *         (or a.length if x is not present).
 */
public static int first(int[] a, int x) {
    int i = 0;

    /* loop invariant:
     *   0 <= i <= a.length
     *   x is not present in the first i elements of a
     */
    while (i < a.length && a[i] != x) {
        i = i+1;
    }
    return i;
}
```

- (a) Suppose in the Netbeans debugger a breakpoint is set on the line starting "while". It will be hit on each iteration just before the condition is evaluated. Suppose somewhere else in the same class the following code is executed.

```
int[] arr = {3, 2, 3, 5};
System.out.println(first(arr, 5));
```

How many times will the breakpoint be hit? Draw a table showing, for each hit, the values of i and $a[i]$. [5%]

- (b) How is a boolean expression $A \ \&\& \ B$ calculated? What would the method `first` do wrong if its while condition were replaced by this one?

```
(a[i] != x && i < a.length)
```

[5%]

- (c) It is now required to write a method `last` that finds the *last* index of x in a (or -1 if x is not present). We decide to use a loop invariant that says

```
/* loop invariant:
 *   0 <= i <= a.length
 *   x is not present in the last i elements of a
 */
```

What are the indexes of the last i elements of a ? Write Java code for `last`, using a while loop, so that this loop invariant holds true on each iteration, immediately before the while condition is evaluated. [5%]

No calculator

4. Consider the following interface.

```
public interface Item {
    public int getPrice();
    public void setPrice(int price);
}
```

Suppose you are also given a public class `BasicItem` that implements `Item`. It uses `getPrice` and `setPrice` as getter and setter for a private field `price`. It also has one constructor, with an `int` parameter used to initialize the price.

- (a) Write Java code to do the following.
- (i) Declare a variable `x` of type `Item`.
 - (ii) Assign to `x` a newly created instance of `BasicItem` with price 25.
 - (iii) Use `getPrice` and `setPrice` to increase the price of `x` by 5. (Note - don't simply set the price to 30.)
- [4%]
- (b) Write Java code for a class `DiscountedItem` that is a subclass of `BasicItem`. It should have the following features.
- (i) It has an extra private integer field called `discount`, with getter and setter `getDiscount` and `setDiscount`.
 - (ii) Its only constructor has two integer parameters, used to initialize price and discount.
 - (iii) It overrides `getPrice`, so that its result is `price*(100-discount)/100`. (Remember that `price` is private!)
 - (iv) It inherits `setPrice` from `BasicItem`.
- [8%]
- (c) Consider the following code.
- ```
BasicItem x = new DiscountedItem(100, 5);
System.out.println(x.getPrice());
```
- (i) What is printed out, 100 or 95? Explain your answer. [1%]
  - (ii) Why would `x.getDiscount()` be illegal? How would you use *casting* to make it legal? [2%]

No calculator

5. Consider the following Java definition of an interface.

```
public interface Applier {
 public double apply(double x);
}
```

This question explores the following idea. If  $f$  is a mathematical function, then we can try to program it in Java as an instance of a class implementing `Applier`. The class defines its method `apply` as a particular way to calculate functions, and then the instance may contain extra parameters special to  $f$ .

- (a) Write a class `Polynomial`, implementing `Applier`, whose `apply` method calculates polynomial functions

$$f(x) = c_0 + c_1x + \dots + c_{n-1}x^{n-1}$$

from their list of coefficients. The coefficients are stored in an instance field `c`, of type array of double, so that the mathematical  $c_i$  is stored as `c[i]`. The array `c` should be initialized from a constructor parameter. You will be given extra marks for clear comments including method headers and loop invariants.

[5%]

- (b) If you differentiate the polynomial  $f(x)$  you get another polynomial  $f'(x)$ , often called the derivative of  $f$ , calculated by

$$f'(x) = c_1 + 2c_2x + \dots + (n-1)c_{n-1}x^{n-2}$$

Write a non-static method `derivative()` in `Polynomial`, whose result is a new `Polynomial` instance for the derivative of this.

[5%]

- (c) The Newton-Raphson method is used to solve equations  $f(x) = 0$ . It says that if  $x$  is an approximate solution, then a better solution is normally given by

$$x - f(x)/f'(x)$$

where  $f'(x)$  is the derivative of  $f$ .

Write a class `NewtonRaphson`, implementing `Applier`, whose `apply` method takes  $x$  and returns the better approximation according to the above formula.  $f$  and  $f'$  should be stored in two private fields `f` and `fDash` of type `Applier` and initialized from two constructor parameters. *Note* - the class does not have to differentiate at all. It just assumes that the second constructor parameter is the derivative of the first.

[5%]

No calculator

**Section B**

6. (a) The following is a listing of the interface class `ActionListener` from the Java API:

```
package java.awt.event;

import java.util.EventListener;

public interface ActionListener extends EventListener {

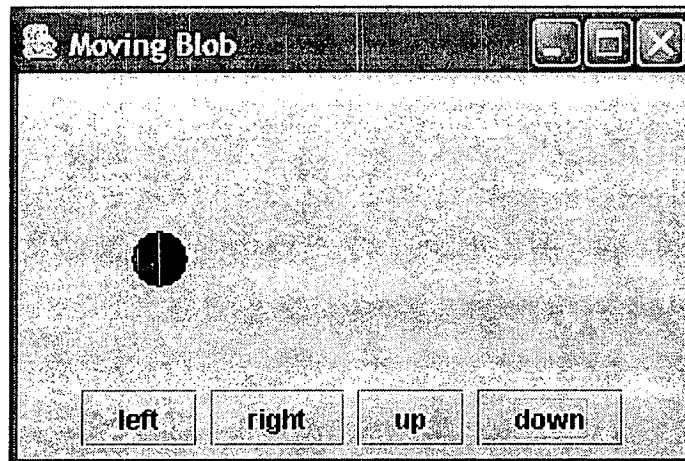
 public void actionPerformed(ActionEvent e);

}
```

Explain the role of this interface in GUI programs written in Java.

[4%]

- (b) Illustrate your answer to (a) by writing classes, using Swing components, that create a window like the following:



When the window is first created, the black blob should start at the top left of the available space. When the left button is clicked, the blob should move to the left by ten pixels, when the right button is clicked, the blob should move to the right by ten pixels. The up and down buttons should move the blob up or down by the same amount. A button should have no effect if the result would be to move part of the blob off the region in which it is being moved. If the window is resized, the region available for moving the blob should change accordingly.

*NOTE: You need not write the code for all four buttons in full - code for implementing one button should be presented together with an indication of where similar code for the other three buttons would be inserted.*

[8%]

No calculator

- (c) Explain the differences between the following two fragments of code for outputting an int to a file

(i)

```
int i;
PrintWriter outfile = new PrintWriter(
 new FileWriter("ints.txt"));
outfile.print(i);
```

(ii)

```
DataOutputStream out = new DataOutputStream(
 new FileOutputStream("ints.dat"));
out.writeInt(i);
```

[4%]



No calculator

7. A Java application processes a Video Shop's video records. The class `Video` has been declared as follows:

```
public class Video
{
 private String title;
 private int reference;

 public Video(int r, String t)
 {
 reference = r;
 title = t;
 }

 public void setReference(int r)
 { reference = r; }

 public int getReference()
 { return reference; }

 public void setTitle(String t)
 { title = t; }

 public String getTitle()
 { return title; }

 public String toString()
 { return reference + ":" + title; }
}
```

- (a) A Java class is required for representing a list of Videos. Write a class called `VideoList` that uses a *linked list* data structure to store a list of Videos. Your class should have a constructor for initialising an empty video list, a method for adding a new `Video` object to the list and a `toString` method that produces a `String` representation of the video list, each video being listed on a separate line. The linked list can be stored in any convenient order. Do *not* use any of the Java Collection classes in your answer to this part. [8%]
- (b) Sketch a diagram showing the pointer structure created as a result of obeying the following code:

```
VideoList vids = new VideoList();
vids .add(new Video(123, "Gone with the Wind"));
vids .add(new Video(456, "Casablanca"));
vids .add(new Video(789, "Dr No"));
```

[4%]

- (c) Demonstrate how the class `Treeset` from the Java Collections Framework could be used in implementing an alternative version of `VideoList` in which the videos are stored in alphabetical order of title. [5%]