

UNIVERSITY OF LONDON

GOLDSMITHS COLLEGE

B. Sc. Examination 2002

COMPUTING

IS52006A (CIS212) Programming: Advanced
Topics and Techniques

Duration: 3 hours

Date and time: May 2002

Do not attempt more than SIX questions on this paper.

You must answer THREE questions from Section A and THREE questions from Section B.

Full marks will be awarded for complete answers to SIX questions.

Each question carries 25 marks. There are 150 marks available for full answers to SIX questions on this paper.

**THIS EXAMINATION PAPER MUST NOT
BE REMOVED FROM THE
EXAMINATION ROOM**

Section A: Object oriented programming in Java

Question 1

A company sells cards of various types eg: birthday cards, postcards, Christmas cards etc. The different types of cards are modeled by a hierarchical class structure.

The top level class `Card` has four instance fields: a real field `price`, two string fields `supp`, `desc` giving the supplier and a description of the card, and an integer field `quant` giving the quantity. The instance method `void details()`, with no parameters, prints out details of an object of type `Card`.

Sub-classes of `Card` include `PostCard` and `Greetings`. The class `Greetings` also has sub-classes `Birthday` and `Seasonal`. All sub-classes except `Birthday` have no extra fields or methods.

The class `Birthday` has two extra instance fields: an integer field `age`, and a boolean field `fun` which says if the card is humorous. The class has its own version of the void method `display()` which, after calling the method `display()` of class `Card`, also prints the age group and humor category of the card.

- (a) Draw a diagram to show the inheritance structure and class members of `Card` and its sub-classes. [5]
- (b) Write the Java code for the classes `Card`, `PostCard`, `Greetings`, `Birthday` and `Seasonal`. [8]
- (c) Use the class `Card` and its sub-classes to illustrate the concepts of method over-riding and upcasting (promotion) by giving suitable examples. [5]
- (d) Write a testbed called `TestCard` which creates an object called `xmas` of type `Seasonal`. Initialize the variables of `xmas` for the following data. The card design is "Holly", supplier is "Oxfam", price is £1.99 and the quantity in stock is 500. [4]
- (e) What would be the result of including the following lines of code for the (un-initialized) object `mine` in the testbed?

```
Birthday mine = new Birthday();  
mine.details();
```

[3]

Question 2

- (a) Explain what is meant in Java by the signature of a method. Illustrate your answer with suitable examples. [4]
- (b) Explain the differences between static and instance methods in Java. [5]
- (c) A class `Histogram` has two static methods, `void stars(int n)` and `void printH(int [] A)`. When passed an integer array `A` as a parameter, `printH()` prints a line of stars, for each row `i` of `A`, to show the value of the entry `A[i]`. The line of stars is printed using method `stars()`. You may assume that the entries `A[i]` are always non-negative.

Write the class `Histogram`, and a class `PrintHist` which makes a call to `Histogram` to print the array with entries `{3, 4, 5, 2, 7}`, and hence produces the output

```
***
****
*****
**
*****
```

[11]

- (d) Discuss the code given below. What would be the output, if any? Is it correct, and if so why, and if not why not?

```
class Person{ String name;
static Person naughty(Person p){
    p.name="Sam";
    return p;}
}

class Persnal{
public static void main(String [] arrgg){
    Person bob = new Person();
    bob.name="Bob";
    bob=Person.naughty(bob);
    System.out.println(bob.name);
}}
```

[5]

Question 3

- (a) Give a brief description of the basic ideas underlying Object Oriented Programming (OOP), and contrast the OOP approach with other styles of programming. [6]
- (b) (i) Explain what is meant by the Java Applications Programming Interface (API)
- (ii) How is it possible to make a method of the API available within a Java program?
- (iii) Suppose a uses a method called `JavaMeth()` from a class in the API. Explain how to obtain the documentation and signature for this method. [4]
- (c) The class `Random` of `java.util` has an instance method `int nextInt()` which takes no parameters and returns an integer.
- (i) Write a class `ATest` which makes a call to this method and prints out the result of this call. The class should also test whether the result of the method call was odd or even. **Note:** The Java remainder operator `%` finds the remainder on integer division, for example `5%2` is 1.
- (ii) How does this example support the underlying philosophy of OOP? [9]
- (d) Explain briefly the significance of the following code fragments
- (i) `public static void main(String [] txt) throws IOException`
- (ii) `class MyFrame extends JFrame` [6]

Question 4

- (a) Explain the term *method overloading* in the context of Java programming. Give an example of a common overloaded method. What is the purpose of method overloading? [4]

- (b) A class `Whatever` has a void overloaded instance method `phrase(param)`. If the parameter is an integer n , it prints out `Youre welcome` n times. If the parameter is a string s , it prints out the string `Youre welcome+s`. If the parameters are two strings s, t , it prints out the string `s+says 'Youre welcome+t'`. Write the class `Whatever`, and suitable method calls to produce the following output:
`Youre welcome`
`Youre welcome`
`Youre welcome`
`Youre welcome Mickey`
`Donald says 'Youre welcome Mickey'` [12]

- (c) The listing of class `Car` is given below.

```
class Car{
    String make; double price; String regNo;
    static int NumCars=0;
    void whatCar( ){
        System.out.println("I am a "+make+" registration number "+
                           regNo+" my price is "+price);}}
```

- (i) Give a brief explanation of the function of class `Car`.
(ii) Write a class called `CarWorld` which creates suitable objects of class `Car` to produce the following output by initializing all variables directly in the main method.

```
I am a BMW registration number THX 208 M my price is 230
There are 1 cars in the showroom
```

- (iii) Write an overloaded constructor for `Car`. Both versions of the constructor increment `NumCars` by 1. The first version of the constructor does not initialize any instance variables. The second version initializes all instance variables. [9]

Question 5

- (a) Explain the method by which graphics components are attached to a top level window, such as `JFrame`, in a Java `Swing` graphical user interface. [5]
- (b) Explain the method by which events can be registered and acted upon in the context of active buttons in a Java graphical user interface. [5]
- (c) The graphical user interface for a simple program is shown below.

The user enters the distance in the first field, and the price, calculated as $\pounds 30 * \text{distance}$, is displayed in the second field when the `Cost` button is clicked.

For example, a distance of 200 has a price of $30 * 200 = \pounds 6000$.

When the `Exit` button is clicked, the program terminates.

.... Insert graphics here (see Overleaf for diagram)....

Write a Java program using `Swing` graphics to produce this graphical user interface. [15]