

R266/304

DUBLIN INSTITUTE OF TECHNOLOGY
KEVIN STREET, DUBLIN 8

DIPLOMA IN COMPUTER SCIENCE

YEAR 3

SUPPLEMENTAL EXAMINATION 2004

PROGRAMMING

MR. C. O'LEARY
DR. B. O'SHEA
DR. C. MULVIHILL

THREE HOURS

ATTEMPT ANY **FOUR** QUESTIONS

ALL QUESTIONS CARRY EQUAL MARKS.

1. (a) With reference to *Java Bytecode*, *classpath* and the *Java Virtual Machine*, explain clearly the compilation and execution processes for a Java program, contrasting it with the compilation and execution of programs written in different languages for native environments.

(6 marks)

- (b) In terms of the Java programming language, and object oriented programming in general, provide a clear definition of the following terms:
- i. Class
 - ii. Object
 - iii. Method
 - iv. Class Member
 - v. Instance Member
 - vi. Constructor
 - vii. Exception

Use sample code to help illustrate your answer.

(10 marks)

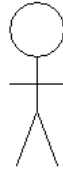
- (c) Java provides a facility whereby related classes can be grouped together into *packages*.
- i. Show how a class can be included in a package.
 - ii. Show how individual classes, contained in a package can be used by another class.
 - iii. Show how a set of classes, in the same package can be used by another class.
 - iv. Identify *two* standard packages that are available in any Java installation, and describe the general function of their constituent classes.
 - v. Describe what is meant by a *Java Archive*, or JAR, demonstrating how a JAR file can be created and used.

(9 marks)

2. (a) With reference to the Java Abstract Window Toolkit (AWT), explain how a Graphical User Interface can be built. In your answer you should identify the roles of *containers*, *components*, *layout managers* and *event handlers*, giving examples of each.

(7 marks)

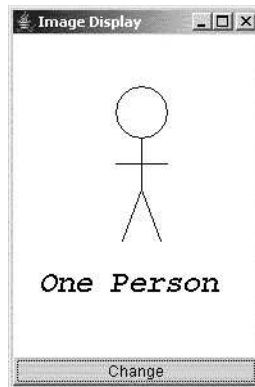
- (b) Create a Java class called `PersonImage` that extends the `Canvas` class and draws the image shown below.



One Person

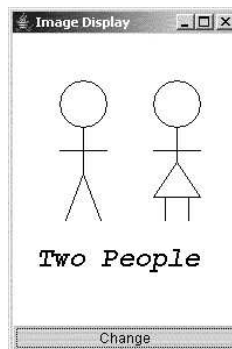
(6 marks)

- (c) Provide the Java code that can be used to put the code from part (b) above into a frame alongside a button, as shown in the image below.



(6 marks)

- (d) Show how the code from parts (a) and (b) above would need to be changed in order to allow the following behaviour:
- When the “*Change*” button is clicked the image changes to *Two People*, as shown below.
 - When it is clicked a second time it changes back to the original image.



(6 marks)

3. (a) Define the term *design pattern*, and discuss why using design patterns aids the development of reusable software. (5 marks)

(b) Using an example and sample Java code, demonstrate the use of the *Factory* design pattern. (7 marks)

(c) Using your example from part (b) above, explain clearly what is meant by the term *polymorphism*. (6 marks)

(d)

```
class A {
    public String toString() { return "A"; }
}
class B extends A {}
class C extends B {
    public String toString() { return "C"; }
}
class D {
    public String toString() {
        return super.toString();
    }
}
public class Main {
    public static void main(String args[]) {
        Object o1 = new A();
        Object o2 = new B();
        Object o3 = new C();
        Object o4 = new D();
        System.out.println(o1);
        System.out.println(o2);
        System.out.println(o3);
        System.out.println(o4);
    }
}
```

Consider the above classes. What output would you expect when these classes are compiled, and `Main` is run. Explain *precisely* why you would expect this output. In your answer you should identify the significance of the `java.lang.Object` class.

(7 marks)

4. (a) Applets are considered *mobile* applications. Discuss why this is the case, identifying clearly how applets differ from standard applications, both in terms of their relationship with the network, and their interaction with their environment. (7 marks)

- (b) Provide the Java and HTML code necessary to create and display in a browser the applet shown below.



(6 marks)

- (c) Show the changes to the code from part (b) above that are necessary so that the applet reads the following information from the source code of the web page:
- The string to be displayed
 - The x co-ordinate of the string's location in the applet
 - The y co-ordinate of the string's location in the applet

(6 marks)

- (d) Show how it is possible to write a Java application that can mimic the function of a browser, by sending a HTTP GET request to a remote web server, requesting it for a web page.

(6 marks)

Continued Overleaf.

5. Consider the following scenario:

- A `SportsPlayer` has a name and an age. `SportsPlayers` fall into two categories, `AmateurPlayers` and `ProfessionalPlayers`.
- An `AmateurPlayer` has a job. An `AmateurPlayer`'s income from sport is 0.
- A `ProfessionalPlayer` has endorsements (money received from advertisements). A `ProfessionalPlayer`'s income from sport is endorsements plus normal income, calculated differently for each type of player.
- A `SoccerPlayer` is a `ProfessionalPlayer`. A `SoccerPlayer` has a weekly wage, and has played for a certain number of weeks. A `SoccerPlayer`'s normal income is the product of weekly wage and the number of weeks played.
- A `RugbyPlayer` is a `ProfessionalPlayer`. A `RugbyPlayer` has a match fee, and has played a certain number of matches. A `RugbyPlayer`'s normal income is the product of the match fee and the number of matches has played.
- A `HurlingPlayer` is an `AmateurPlayer`.
- A `GaelicFootballPlayer` is an `AmateurPlayer`.
- `HurlingPlayers` and `RugbyPlayers` play dangerous sports, and hence both are also `InsuredPlayers`.
- `InsuredPlayers` have an insurance policy of a certain value.

(a) Define the terms abstract class and interface, with reference to the Java programming language. Distinguish clearly between the two.

(4 marks)

(b) Provide the Java code necessary for each of the classes/interfaces required for the scenario described above.

All `SportsPlayers` should support a method called `getIncomeFromSport()`, which should return the total income of the `SportsPlayer`. All `InsuredPlayers` should support a method named `getInsurancePolicyValue()` which should return the amount of their insurance policy.

(15 marks)

(c) Provide the Java code for a simple program control class that should do the following:

- i. Creates four different `SportsPlayers`, one of each type, and stores them in an array.
- ii. Iterates through the array, calculating the total income of all the players together, and the total amount for which all the players are insured.
- iii. Displays both values calculated in part (ii) above.

(6 marks)

6. (a) Explain clearly what is meant by the terms *information hiding* and *encapsulation*. Use an example to illustrate your answer. **(4 marks)**
- (b) The `this` keyword in Java can be used as a reference value, and as a method name. Using sample code, explain clearly how it is used in each of these two ways, and what purpose it serves. **(5 marks)**
- (c) Identify four different levels of access that can be applied to instance members of a class. Distinguish clearly between each of these access levels, using clear examples to help illustrate your answer. **(5 marks)**
- (d) In relation to Java class methods, define the term *overloading*. **(3 marks)**
- (e) An `Employee` has an identification number, a name and a department.
A `Department` has a name and a boss, who is also an `Employee`.
Design the `Employee` and `Department` classes, employing the principles identified and discussed in parts (a) to (d) above. **(8 marks)**