

OLLSCOIL NA hÉIREANN, GAILLIMH
THE NATIONAL UNIVERSITY OF IRELAND, GALWAY

SEMESTER 1 EXAMINATIONS 2000/2001

B.A. Degree Examination
B.A. (International) Degree Examination

CT326: PROGRAMMING III

Prof. D. Bell
Prof. G. Lyons
Dr. M. Madden

Time Allowed: **Three Hours**
Answer Any **Four** Questions
All Questions Carry **Equal** Marks

1. (a) Write the code for a Java class called **Radio**, with appropriate member variables for keeping track of whether the radio is on or off, the current station (a number in the range 1 to 100) and its volume (a floating-point number in the range 0 to 10). Include a constructor for the class. The public interface of the class must include methods **on**, **off**, **setStation** and **setVolume**. Each of these methods must change the member variables as appropriate. The methods **setStation** and **setVolume** should validate the data passed to them.
Note: You do not need to display any messages or have other output from any of these methods. (10)
- (b) For the **Radio** class, add a public method called **getState**, which returns a text string describing the current state of the radio, e.g. "On; Volume = 5; Station = 2". (3)
- (c) Based on the **Radio** class from Part (a) above, derive a class called **StereoRadio**, which is a radio with the addition of balance (a number in the range -1 to 1) and extra bass (which can be on or off). Include a constructor for the class, and additional public methods called **bassOn**, **bassOff**, and **setBalance**; these must work similarly to the methods defined in the **Radio** class. (9)
- (d) Write a **getState** method for the **StereoRadio** class, which returns a text string describing the state of the stereo radio. (3)

2. (a) Describe Java's six Boolean operators, distinguishing clearly between operators that are similar. (9)
- (b) Write the Java code to declare each the following variables, and set their initial values:

Name	Type	Value
i	integer	57
f	floating-point number	2.22
d	double-precision number	8000
s	text string	Hello

(4)

- (c) Using the variables declared in Part (b) above, explain using lines of code how to convert data:

- (1) from integer to text string
- (2) from integer to floating-point number
- (3) from floating-point number to integer
- (4) from text string to floating-point number
- (5) from double-precision number to text string
- (6) from floating-point number to double-precision number

(12)

3. (a) State the general format of a class definition in Java. Explain the meaning of all of the keywords, and indicate which keywords are optional. (6)

- (b) Explain each of the following terms:

- (1) Encapsulation
- (2) Inheritance
- (3) Polymorphism

(9)

- (c) Write a Java program that asks the user to input three numbers (**doubles**), and displays the minimum, maximum, sum, average and midpoint. Each of these quantities should be calculated in a separate method. Each method should take three doubles as input parameters and returns the answer as a double. The inputting and display should all be done in the **main** method.

Note that the midpoint of a set of numbers is found by adding the minimum and the maximum, and dividing the answer by two. Also, the average of three numbers is the sum of those numbers divided by three. Methods should call each other, where appropriate, to avoid repeated code.

(10)

4. (a) Explain the term **structured programming**, listing its four key features. (4)
- (b) Describe each of the key features of structured programming that you have listed above, and explain the different Java structures that implement each of them. (12)
- (c) Java has three repetition structures. For each of these, give an example of a situation where it would be the most appropriate option. (9)

5. (a) Explain each of the following Java programming terms, using short code samples to illustrate your answer:
- (1) Method overloading
 - (2) Static methods
 - (3) Constants (12)
- (b) Write a Java program that repeatedly asks the user to enter an integer number, and keeps track of the number of even and odd numbers entered. It must stop when the user enters 0 (zero), and display the total count of even numbers and the total count of odd numbers. The program must include the following methods:
- (1) A method called **isEven**, that takes an **int** parameter and returns **true** if the **int** is even, or **false** if it is not.
Note: if a number is even, when it is divided by 2 the remainder is zero.
 - (2) A method called **getIntegerFromUser**, that takes no parameters and returns an **int**. The method must use an input dialog ask the user to enter a number, and return the value entered by the user.
 - (3) A **main** method, with the rest of the functionality required for the program to work. (13)
6. Ahascra International Bank is implementing an Internet Banking System. Your responsibility is to develop the security validation system that people will use to log on.
- For each customer, the security data stored consists of: **firstName** (text string); **lastName** (text string); **accountNo** (long integer); **password** (text string). You can assume the password is always 6 letters long.
- (a) Write a Java class called **SecurityData**, with four private members as listed above. (2)
 - (b) For the **SecurityData** class, implement the following public methods:
 - (1) **getName**: returns the first and last name joined together, with a space in between, e.g. "Thomas Dempsey".
 - (2) **getAccountNo**: returns the account number.
 - (3) a constructor, which has parameters for specifying the first name, the last name, and the account number, and which sets a default password consisting of the first five letters of the customer's last name and the number of letters in the customer's first name (e.g. "Demps6").

Note: If **s** is a **String**, then : **s.length()** returns the number of letters in the string, and **s.substring(0,3)** returns the first three letters of the string. (9)
 - (c) For the class, write a method called **checkPassword()** to check the user's password. This is done by repeating the following procedure three times: Randomly select a position in the password, ask the user for the letter at that position (e.g. "Enter the letter at Position 3 in your password"), and verify that the letter entered is correct. If all details entered are correct, display a message saying "Welcome". Otherwise, display a message saying "Access Denied".
- Note: **Math.random()** returns a random number between 0 and 1; and for a **String s**, **s.charAt(3)** returns the character at position 3 of the string, counting from 0.) (14)