GX1583

OLLSCOIL NA hÉIREANN, GAILLIMH

THE NATIONAL UNIVERSITY OF IRELAND, GALWAY

SEMESTER 1 EXAMINATIONS 2002/2003

B.A. Degree Examination

B.A. (International) Degree Examination

CT335: 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) List the Java constructs for performing decisions. For each one, explain its format and give an example of a situation where it would be the most appropriate option. (7)

   (b) Unlike other programming languages, Java has several assignment operators; explain what the point of this is. List six assignment operators, describing what each one does and giving an example of its use. (7)

   (c) Examine the following Java code and write out the output it produces. (Note: for clarity, use a symbol such as "_" to indicate a space output by the program.)

```
int i, j;
for (i=4; i>=1; i--)
{
   for (j=i; j>=1; j--)
   System.out.print(" ");
   while (j++ < 9-2*i)
      System.out.print("*");
   System.out.println();
}
for(j=10; j<15; i=0, j+=3)
{
   while (i<5)
      System.out.print(++i==5 ? "H" : " ");
   System.out.println();
}
```

   (10)

   (d) What are the values of `i` and `j` after this code has been executed? (1)

2. (a) EntertainMe.com sells two types of product, movies that are rented and music that is sold. You are required to develop code to keep track of sales/rentals.

Write a class called **Product**, with members to hold a title and a product ID. In the class definition, ensure that it will not be possible to create objects of this class. Include the following public methods:

- A constructor with parameters for specifying the title and ID
- Access methods that return the title and ID
- A method with no definition called **getEarnings,** which returns a floating-point number, and which will be defined in all classes based on the **Product** class
- A method called display that displays a message on screen with the product's title and its earnings.

(11)

(b) Write a class called **Movie**, based on **Product**. Include member variables for holding the movie rental price and the number of rentals. Your code must account for the fact that the movie rental price is the same for all movies, but the number of rentals is stored individually for each movie. Include the following public methods:

- A constructor with parameters for specifying the movie title and ID, that initialises the number of rentals to 0
- A method for setting the rental price
- A method **newSale** that increments the number of rentals
- A definition of **getEarnings**, that calculates the rental price multiplied by the number of rentals

(8)

(c) Write a class called **Music**, also based on **Product**. Include member variables for holding the music sale price and the number of sales. Again, your code must account for the fact that the sale price is the same for all music tracks. Define a set of methods similar to those defined for the Movie class.

(6)

3. Describe in detail **all** of the following Java concepts, using code examples for illustration where appropriate:

(1) Operator precedence and associativity

(2) Structured programming

(3) Literal constants and named constants

(4) Interface classes

(5) Visibility modifiers

(25)

(a) Message passing is a key characteristic of object-oriented programming languages. How do Java objects send messages to each other? Illustrate with an example. (4)

(b) A software object has state, behaviour and identy. Explain each of these terms, and how they are implemented in Java. (5)

(c) One notable feature of Java is that it is *architecture neutral*. Explain what this means and how it is achieved in the way Java programs are compiled and run. (6)

(d) Write a Java program that prompts the user to enter a sentence, and then displays a message saying how many uppercase letters, lowercase letters and other characters there are in the sentence. You may need to use the following methods:

`s.length()`: returns length of string s
`s.charAt(i)`: returns char at position i in string s
`Character.isUpperCase(c)`: returns true if c is an uppercase char
`Character.isLowerCase(c)`: returns true if c is a lowercase char

(10)

5. (a) Using appropriate examples, discuss the terms *inheritance* and *composition*. During software design, how would you decide whether inheritance or composition would be more appropriate in a given situation? (7)

(b) Discuss how fundamental (simple) variables and objects are handled when they are used as parameters to a method. (6)

(c) Tuam Zoo (formerly the CBS) has employed you to assist with developing an animal record system. Below is a description they have provided:

For all animals, we will store the name, country of origin and birth date. However, all animals are categorised as either Birds, Mammals or Reptiles. For each bird, we will store plumage colour and whether or not it can fly, as well as the details listed above. For both mammals and reptiles, we will store its number of legs and whether it is a carnivore or not, as well as the generic animal details listed above. For mammals, the weight and height will also be stored, whereas for reptiles the length will be stored.

Based on this description, draw a diagram showing an appropriate class hierarchy for the animal data. Using the diagram, develop a **partial** implementation of classes required. You must include all member variables identified in the description, and constructors to initialise them to arbitrary values (e.g Name: "NoName", Origin: "Ireland", etc.), but you do not need to include any other methods. (12)

6. Caymen Islands Internet Bank is employing you to develop a program to assist with their new mortgages business. Below is the code for a the program's main method:

```
public static void main(String[] args)
{
   double loan, repayment;

   repayment = askForPositiveNumber("Enter annual repayment");
   loan = findMaxMortgage();

   JOptionPane.showMessageDialog(null,
      "It will take " + calcYears(loan, repayment)
      + " to repay " + loan + " euro with "
      + " annual repayments of " + repayment + " euro");

   System.exit(0);
}
```

You are required to write the methods to complete this program, to adhere to the following requirements:

(a) `askForPositiveNumber`: Displays an input box with the specified prompt, and returns a number corresponding to the user's input. The method must ensure the number entered is positive (greater than 0). (5)

(b) `findMaxMortgage`: Calculates a maximum mortgage amount based on annual salary and on whether one or two people are applying for the mortgage. For a single person, the maximum is three times the salary, whereas for two people the maximum is 2.5 times the first salary plus 1.25 times the second salary. The method must prompt for details required, ensuring reasonable values are entered (e.g. salaries are positive). (10)

(c) `calcYears`: Calculates the number of years to pay off the mortgage with the given annual repayment, assuming an annual interest rate of 6%. Each year, the annual interest is added to the current balance and the annual repayment is subtracted. If the balance does not decrease, the repayment is too small and the program must take appropriate action. (10)