

Ollscoil na hÉireann, Gaillimh

GX 2080

National University of Ireland, Galway
Semester I Examinations 2005 / 2006

Exam Code(s)	3BA1, 4BA4, 1EM1
Exam(s)	
Module Code(s)	CT335
Module(s)	Object Oriented Programming
Paper No.	1
Repeat Paper	
External Examiner(s)	Prof. Sally McClean
Internal Examiner(s)	Dr. Michael Madden Mr. Jason Byrne

Instructions: Answer any 3 question
All questions will be marked equally

Duration	2 hrs
No. of Pages	5
Department(s)	Information Technology
Course Co-ordinator(s)	

Requirements:

MCQ
Handout
Statistical Tables
Graph Paper
Log Graph Paper
Other Material

CT335: Object Oriented Programming

Answer any 3 question

Time allowed: 2 hours

Question 1

Part A (10 marks)

Write a `GoodbyePrinter.java` class, containing only a main method, that prints out the following using loops:

Hello

Hello

Hello

Hello

Hello

Part B (10 marks)

Explain the static keyword – in doing so, explain the purpose of static data as well as static methods.

Part C (30 marks)

A football team can only have 11 players on the pitch at any time. Write a `Footballer` class that only allows 11 football players to be created. Write a suitably named exception that will be thrown if a user tries to create a `Footballer` and 11 already exist. Inside a main method, show how to try to create 12 `Footballer` objects and catch the resulting exception.

Question 2

Part A (15 marks)

Write a `Bus` class that simulates a bus. A bus can hold and pick up passengers. It can also have its lights turned on and off.

Supply the following methods only:

- A default constructor that takes no parameters. This assumes that there are no passengers on the bus and that the lights are off.
- A second constructor that takes two parameters: an int that indicates how many passengers there are on the bus, and a boolean that indicates if the lights are on or off.
- A “turnLightsOn” method that turns on the bus’s lights.
- A “turnLightsOff” method that turns off the bus’s lights.
- An “areLightsOn” method that returns true if the lights are on; false, otherwise.
- A “getNumPassengers” method that returns the number of passengers on the bus.
- A “pickUpPassenger” method that increases the number of passengers by 1.

Part B (15 marks)

Write a `StudentIDChecker` class that provides a method that takes a student ID (containing numbers and/or characters) and checks to see if it is of length 8.

You may need to use this hint: `s.length()` returns the length of a string `s`.

Part C (20 marks)

Write a mock-up of a piece of web client software (no actual web connection or socket code is required) that allows a user to supply the IP address of the web server and the port number the user wishes to connect to. For example, the user may connect to the web server with IP address 192.168.2.1 over port 80 by running the WebClient class as:

java WebClient 192.168.2.1 80

To simulate the actual connection process, just print a statement containing the IP address and port number. For example, if the command above is successful print the statement:

connecting to 192.168.2.1 on port 80...

Assume that the first argument will be the IP address (or machine name) and that the second argument will be the port number.

Write some error correcting code that checks if the user has entered the right amount of arguments and return different helpful error messages if the user has entered:

- No arguments
- One argument
- More than two arguments

Also check that the port number is a valid integer and not some random String by trying to parse the second argument using

Integer.parseInt(int argument)

If the port number is not a valid integer, print a helpful error message to the user and exit the program. Beware that **parseInt** may throw a **NumberFormatException**.

Question 3

Part A (20 marks)

Imagine you have a knapsack that can only hold a fixed volume of camping gear – say 20 units. Inside a main method, write code that will calculate the heaviest object of increasing weight we can put in the knapsack – we start with an object of weight 1, then add an object of weight 2, then 3, and so on. How heavy is the last object you can put in without going over the 20-unit limit? Use a while loop to calculate this – do not just add 1, 2, 3, ...

Part B (10 marks)

The following piece of code is supposed to print a series of indented asterisks “*” to screen like so:

```
*
*
*
*
*
```

Unfortunately, there are three mistakes that will prevent it from compiling and running in the desired fashion. Identify them.

```

for (int i = 0, i < 5, i++)
{
    int spaceCounter = i;
    j = 0;
    while (j < spaceCounter)
    {
        System.out.print(" ");
        j++;
    }
    System.out.print("*");
}

```

Part C (20 marks)

Examine the following piece of code and write out the pattern of 'x's and 'o's that is printed to screen.

```

for (int i = 0; i < 9; i++)
{
    if (i % 3 == 0)
    {
        System.out.println();
    }
    if (i % 2 == 0)
    {
        System.out.print("x");
    }
    else
    {
        System.out.print("o");
    }
}

```

Question 4

Part A (15 marks)

To illustrate inheritance, create a Man and a Woman class that extend from a Human class. Make sure that it is not possible to create a Human object by itself. Ensure that each Human has an IQ that is required upon creation and provide a getIQ() method. Also ensure that all types of Humans can say what gender they are.

Part B (20 marks)

Write a CanFly interface declaring a single method, fly(), and a single piece of data representing the speed of sound (300 metres per second).

Show that an interface can be extended by creating a CanFlyReallyFast interface containing an additional method, turboBoost().

Create a BumbleBee and SuperHero class that use the CanFly interface and a Jet that can use the CanFlyReallyFast interface.

Part C (15 marks)

Inside a main method, create a BumbleBee, a SuperHero, and a Jet. Also show how each of them could be put into the same array. Using this array and a **for** loop, call the fly() method on each object.

Question 5**Part A (10 marks)**

Explain what is meant by the **final** keyword. Explain what final data, final methods, and final classes mean.

Part B (20 marks)

Explain each of the following terms:

- IDE
- API
- JVM
- Jar file
- Batch file

Part C (20 marks)

Write an interest calculator class that can prompt the user (with dialog boxes) to enter an initial sum, a rate of interest, and the period of investment. The user is then prompted with a dialog box showing the final sum (calculated using compound interest).