UNIVERSITY *of* LIMERICK

OLLSCOIL LUIMNIGH

COLLEGE *of* INFORMATICS *and* ELECTRONICS
Department of Computer Science and Information Systems

End of Semester Exam

Academic Year:	2007/2008	**Semester:**	Autumn
Module Title:	Imperative Programming 1 Introduction to Computer Programming Programming Language 1	**Module Code:**	CS4411 CS5021 CS5111
Exam Duration:	2½ Hours	**% of Total Marks:**	75
Lecturer:	A. McElligott		

Instructions to Candidates:

Section A: ALL questions should be attempted. 55%

Q1	15%
Q2	12%
Q3	16%
Q4	12%

Section B: ONE question should be attempted. 20%
ALL questions in this section carry equal marks.

State clearly any assumptions you make.

Section A: all questions should be attempted

Q1 **(15 marks)**

What output is produced to screen when

(i) FinalTestDryRun1 (cf. Figure 1.1), **[5 marks]**

(ii) FinalTestDryRun2 (cf. Figure 1.2) and **[5 marks]**

(iii) FinalTestDryRun3 (cf. Figure 1.3) are executed? **[5 marks]**

```
import javax.swing.JOptionPane;
public class FinalTestDryRun1
{
  public static void main(String[] args)
  {
    String r = "";
    for (int i = 0; i < 8; i++)
      if (i % 2 == 0)
        r += (i + 1) + "\n";
      else if (i % 3 == 0)
        r += (i * 1) + "\n";
      else if (i % 5 == 0)
        r += (2 * i - 1) + "\n";
      else
        r = (r + i) + "\n";
    JOptionPane.showMessageDialog(null,r);
  }
}
```

Figure 1.1

```
import javax.swing.JOptionPane;
public class FinalTestDryRun2
{
  public static void main(String[] args)
  {
    short a = 4, b = 0, c;
    String r = "";
    while (b < a)
    {
      c = b;
      while (c < a - 1)
      {
        r = r + "/"; c += 1;
      }
      c = 0;
      while (c < 2 * b + 1)
      {
        r += "*"; c += 1;
      }
      r += "\n";  b += 1;
    }
    JOptionPane.showMessageDialog(null,r);
  }
}
```

Figure 1.2

```
import javax.swing.JOptionPane;
public class FinalTestDryRun3
{
  public static void main(String[] args)
  {
    String word = "introductory programming", r = "";
    int i, j;
    for (i = 1, j = 2; i <= 3; i++, j++)
    {
      r += word.substring(i, ++j) + "\n";
      r += word.indexOf("o", j + i) + "\n";
      r += word.lastIndexOf("r", i + 4) + "\n";
    }
    JOptionPane.showMessageDialog(null,r);
  }
}
```

Figure 1.3

Q2 **(12 marks)**

You are required to write a Java application that will accept a sentence from the end user similar to that shown in Figure 2.1. Your program should capitalize the first character of each word (i.e. convert the first character of each word to uppercase) and display the result (cf. Figure 2.2).

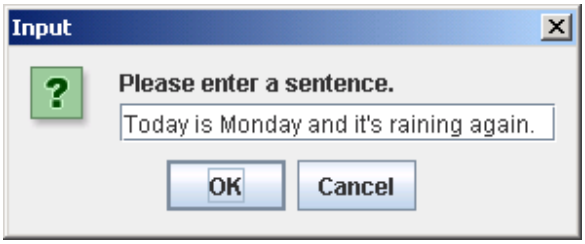

Figure 2.1

Figure 2.2

Q3 **(16 marks)**

The characters A, B, C, D, E, F and G are used to represent notes in traditional music theory. You are required to write a Java application program that will generate random notes. Your program should ask the end user to specify how many notes he/she would like to have generated (cf. Figure 3.1). If the user inputs a number in the desired range your program should display each random note and its frequency (cf. Figure 3.2). An appropriate error message should be displayed (i) if the user fails to supply input (ii) if the user input is not of the correct format or (iii) if the user input is not in the desired range.

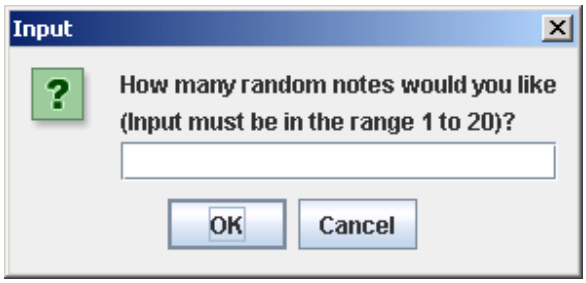

Figure 3.1

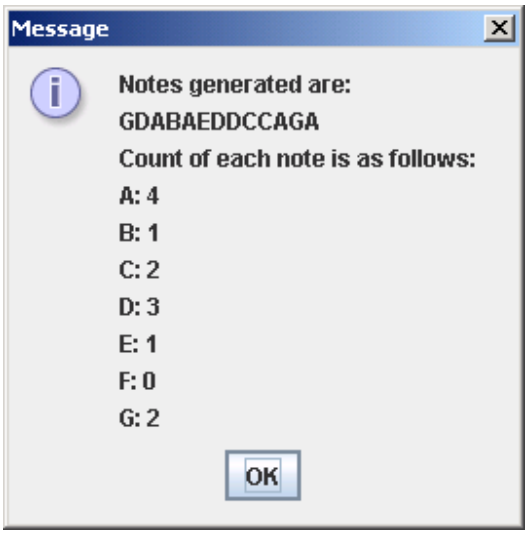

Figure 3.2

Q4 **(12 marks)**

This question relates to the project work you undertook during this semester.
Assume that your program has already generated five unique numbers representing a hand of cards and that the values and suits of each card have been determined and stored in two parallel arrays namely values and suits. You are required to write a method that will accept these parallel arrays and order them so that on completion the card values are in descending sequence. Your program should not continue to pass through the data once it has been determined that the data is already sorted.

End of Section A

Section B: attempt one question

Q5 **(20 marks)**

Each type of credit card begins with a prefix or range of prefixes and is of a certain length. Table 1.1 shows the details of two commonly used credit card types.

Card Type	Prefix(es)	Length
MasterCard	51-55	16
Visa	4	13 or 16

Table 1.1: Details of two commonly used credit card types

In addition to these rules a credit card number is subjected to the Luhn formula to determine if it is valid. The following are the steps of this formula:

1. Take each odd positioned digit and multiply it by 2. If the result of multiplying one such digit by 2 is a 2-digit number these digits should be summed so that the result is a single digit.
2. Add results of all multiplications in Step 1.
3. Sum all even positioned digits.
4. Add the results of Step 2. to the results of Step 3.
5. Modulus the results of Step 4 with 10. If the result is 0 this indicates that the credit card number is valid.

e.g. Credit Card Number 4567123474567836

Step 1: (4 * 2) + (6 * 2) + (1 * 2) + (3 * 2) + (7 * 2) + (5 * 2) + (7 * 2) + (3 * 2)
= 8 + 12 + 2 + 6 + 14 + 10 + 14 + 6
= 8 + 3 + 2 + 6 + 5 + 1 + 5 + 6
Step 2: 36
Step 3: 5 + 7 + 2 + 4 + 4 + 6 + 8 + 6
= 42
Step 4: 36 + 42
= 78
Step 5: 78 modulus 10 = 8 thus invalid

You are required to write a Java application that will accept a credit card number and its type from the end user. Your program is expected to cater for the credit card types shown in Table 1.1. The output from your program should be a message stating whether or not the details entered are valid.

Q6 **(20 marks)**

Vehicle registration plates in Ireland are of the format

Year – Index Mark – Registration Number

where Year is a 2-digit number that refers to the year in which the vehicle is first brought into use, Index Mark refers to the licensing authority area of where the registered owner normally lives, and a registration number is no longer than 7-digits. Each Index Mark corresponds to a Placename. Table 1.2 shows for each Index Mark its corresponding Placename. If a person intends bringing a vehicle into use he/she can reserve a vehicle registration number plate. However, a person cannot reserve a registration number plate if it is the first number of each year in the cities of Cork, Dublin, Limerick and Waterford. For example, for the year 2008, 08 – C – 1, 08 – D – 1, 08 – L – 1 and 08 – W – 1 are reserved for the Mayor/Lord Mayor of each of these cities.

P.T.O.→

Index Mark	Placename	Index Mark	Placename	Index Mark	Placename
CW	Carlow	L	Limerick City	RN	Rosommon
CN	Cavan	LS	Laois	SO	Sligo
CE	Clare	LM	Leitrim	TN	Tipperary North
C	Cork	LK	Limerick County	TS	Tipperary South
D	Dublin	LD	Longford	W	Waterford City
DL	Donegal	LH	Louth	WD	Waterford County
G	Galway	MO	Mayo	WH	Westmeath
KY	Kerry	MH	Meath	WX	Wexford
KE	Kildare	MN	Monaghan	WW	Wicklow
KK	Kilkenny	OY	Offaly		

Table 1.2

You are required to write a Java application program that will allow the end user to check if a vehicle registration number plate can be reserved for the year 2008. This program should accept the data relating to a registration plate i.e., Year – Index Mark – Registration Number as a single data item.

- If the user fails to supply input or if the user input is not of the correct format an appropriate error message should be displayed.
- If the year is invalid an appropriate error message should be displayed.
- If the index mark is invalid an appropriate error message should be displayed.
- If the user enters details of a vehicle registration plate that is reserved for a Mayor/Lord Mayor an appropriate message should be displayed.
- If the user enters details of a vehicle registration plate (that is not reserved for a Mayor/Lord Mayor) your program should display a message stating a valid registration plate has been entered but a check must be performed to see if another person has reserved this registration plate.

End of Section B