XCS10111

UNIVERSITY OF DUBLIN
TRINITY COLLEGE

Faculty of Engineering, Mathematics & Science

School of Computer Science & Statistics

Integrated Computer Science Programme
B.A. (Mod.) Computer Science Linguistics & Language
B.A. (Mod.) Business & Computing
B.A. (Mod.) Management Science & Information Systems Studies
Junior Freshman Annual Examination

Trinity Term 2012

CS1011 Introduction to Programming I & II

Thursday, 10th May 2012 LUCE UPPER 14:00 – 17:00

Dr. Stephen Barrett, Dr. Arthur Hughes and Dr. Kenneth Dawson-Howe

Instructions to Candidates:

Attempt **four** questions.
All questions carry equal marks.
Answer each question in a separate answer book.

Materials permitted for this examination:

None required.

Q1. *Animal, Vegetable, Mineral* is a classic computer game where the player thinks of an object, and the computer attempts to guess the object by asking a series of binary questions, such as:

+++ Are you thinking of an Animal? +++
yes

Depending on the answer, the computer will ask further questions, seeking to narrow down the range of possible answers to a single answer:

+++ Does it have fur? +++
yes
+++ Does it have a tail? +++
yes
+++ Is it a monkey? +++
no

Eventually the program will run out of questions and will either be able to guess the object correctly, or it will be unable to guess the object.

Write a Java application that can play this game. You solution must correctly guess the answer in the fewest possible number of steps. Your solution should store a pre-existing data set regarding questions to ask. You need not provide a solution capable of learning.

Q2. A *palindromic* word is a word that can be read the same way in either direction. Examples of palindromic words include:

Anna
tattarrattat

Write a Java program capable of identifying, and counting, palindromes in a text file (ascii) presented as input. The name of the file should be accepted as command line input.

Q3. The game *higher-lower* is played with 5 random cards selected from a pack and placed in a row, face down. The first card is turned over, and the player has to guess whether the next card is higher or lower than the current card. The game proceeds until either the player has guess incorrectly, or the player has correctly guessed for the final card. Should any two cards be the same, this counts as an incorrect guess.

Write a High-Low application in Java in which a user has to try to choose whether a series of playing cards are higher or lower than the current card. You may assume the existence of a method in a base class that returns a set of 5 valid random cards, with the signature `String[] getCards()`.

Q4. Write a `Date` class whose instances represent dates. Your class should provide the following methods:

(i) an `isLeapYear` method which determines whether the passed year is a leap year or not. If the year is divisible by 4 it is a leap year, unless it is divisible by 100 in which case it is not, unless it is divisible by 400 in which case it is a leap year;

(ii) an `isValidDate` method which checks to see if a passed year, month and day combination are valid;

(iii) a constructor which takes the Year, Month and Day as parameters and which raises an exception if the date is not valid.

Q5. A *Substitution Cipher* replaces plaintext with ciphertext. The most common substitution ciphers replace single characters of plaintext with predefined single characters of ciphertext (e.g. the plaintext character 'a' might be replaced by cipher text character 'q', 'b' might be replaced by 'x', 'c' by 'k' and so on). Each plaintext character should be replaced by a different ciphertext character.

Write a Java class `SubstitutionCipher` whose instances represent substitution ciphers. Your class should provide the following methods:

(i) a constructor method which takes a list of 26 ciphertext characters corresponding to the plaintext characters 'a' .. 'z';

(ii) a constructor method which randomly assigned ciphertext characters to each plaintext character 'a' .. 'z';

(iii) an `encrypt` method which takes a string and returns an encrypted version of the string according to the substitution cypher;

(iv) a `decrypt` method which an encrypted string and returns an decrypted version of the string according to the substitution cipher.

Q6. Write a Java class `Matrix` whose instances represent 2D matrices of `doubles`. Your class should provide the following methods:

(i) a constructor method which takes a 2D array of `doubles` and initialises the matrix from this;

(ii) an `averageValue` method which returns the average value of the cells in the matrix e.g. if the matrix is $\begin{pmatrix} 2.0 & 6.0 \\ 8.0 & 4.0 \end{pmatrix}$ then the average value is 5.0;

(iii) a `normalise` method which returns a new matrix where the sum of the elements in the matrix is 1.0 e.g. if the matrix is $\begin{pmatrix} 2.0 & 6.0 \\ 8.0 & 4.0 \end{pmatrix}$ then the normalised matrix is $\begin{pmatrix} 0.1 & 0.3 \\ 0.4 & 0.2 \end{pmatrix}$;

(iv) an add method which takes two matrix objects and returns a new matrix object whose cells are the sum of the corresponding cells in the passed matrix objects e.g.

$$\begin{pmatrix} 0.1 & 0.3 \\ 0.4 & 0.2 \end{pmatrix} + \begin{pmatrix} 0.5 & 0.2 \\ 0.7 & 0.01 \end{pmatrix} = \begin{pmatrix} 0.6 & 0.5 \\ 1.1 & 0.21 \end{pmatrix}.$$