PRINCE OF SONGKLA UNIVERSITY

FACULTY OF ENGINEERING

Department of Computer Engineering

Midterm Examination: Semester 2 **Academic Year**: 2011-2012

Date: 19th December, 2011 **Time**: 13:30 – **15:30 (2 hours)**

Subject Number: 241-211 **Rooms**: R200, S203,

Subject Title: Object Oriented Programming Robot Head

Lecturer: Aj. Andrew Davison

Exam Duration: 2 hours

This paper has 4 pages.

Authorized Materials:

- Writing instruments (e.g. pens, pencils).
- Books (e.g. dictionaries) and calculators are **not** permitted.

Instructions to Students:

- *Answer questions in English.* Perfect English is **not** required.
- Attempt all questions.
- Write your answers in an answer book.
- Start your answer to each question on a new page
- Clearly number your answers.
- Any unreadable parts will be considered wrong.
- When writing programs, use good layout, and short comments; marks will not be deducted for minor syntax errors.
- The marks for each part of a question are given in brackets (...).

Question 1

(30 marks; 30 minutes)

a) Explain the differences between a *class* and an *object*? (13)

b) Explain *call-by-value* and *call-by-reference* parameter passing in Java. (12)

c) What is *modularization* and *abstraction*? (5)

Explain using words, diagrams, and small code fragments in your answers.

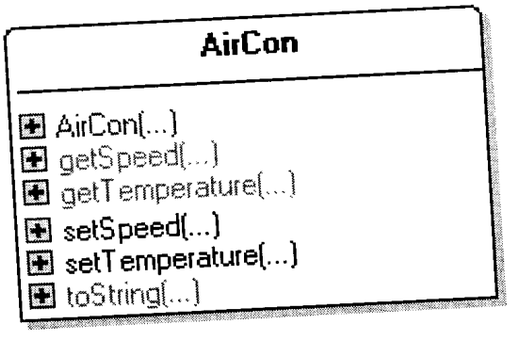
Question 2

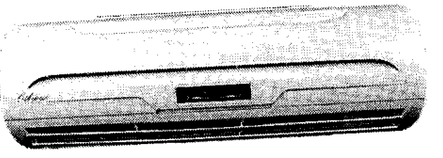
(20 marks; 20 minutes)

a) Write a Java class called `AirCon` to model an air-conditioner. The public methods are shown in the class diagram below (but the private data is hidden). You must implement the class.

An air-conditioner has a speed setting (which can be set to low, medium, or fast), and a temperature setting (which can only vary between 18 and 30 degrees Celsius). The toString() method returns a string consisting of all the data in an object.

Your should implement the Speed type (required by getSpeed() and setSpeed()) as a Java enum. (14)





b) Write a short `TestAirCons` class that shows how **two** `AirCon` objects can be created and used. Draw *diagrams* explaining how the objects work. (6)

Question 3

(30 marks; 30 minutes)

The following main() function implements the top-level of a Hangman game.

```
// globals
private static final int MAX_TRIES = 8;

private static String[] words = {
  "acorn", "actor", "album", "alien", "bagel",
  // ... lots more words, all in lowercase
};

public static void main(String[] args)
{
```

```
ArrayList<Character> guessedLetters = new ArrayList<Character>();
Random rand = new Random();
Scanner in = new Scanner(System.in);

String word = words[ rand.nextInt(words.length) ];
boolean isGameOver = false;
int triesLeft = MAX_TRIES;

while (!isGameOver) {
  printGuesses(guessedLetters);
  System.out.println("You have " + triesLeft +
                     " tries left to find the word: ");

  System.out.println("    " +
          maskWord(word, guessedLetters) + "\n");

  char letter = makeGuess(in, guessedLetters);
  if (word.indexOf(letter) == -1) {
    System.out.println("  Sorry, " + letter +
                       " is not in the word.");

    triesLeft--;
    if (triesLeft == 0) {
      System.out.println("HANGMAN! The word was: " + word);
      isGameOver = true;
    }
  }
  else {
    System.out.println("  Good, " + letter + " is in the word");
    if (word.equals( maskWord(word, guessedLetters) )) {
      System.out.println("You got it! The word was: " + word);
      isGameOver = true;
    }
  }
 }
} // end of main()
```

a) Explain <u>in words</u> how the Hangman game works. Do **not** write out the entire main() function in your answer. (5)

b) Implement **maskWord**() using the StringBuilder class. Document your code. *Hint*: use ArrayList.contains() to test if a character is in a list. (5)

c) Explain the differences between the String and StringBuilder classes. (4)

d) Implement **printGuesses**() using a for-each loop. Document your code. (4)

e) Explain the differences between a for-each loop and a standard for loop. (4)

f) Implement **makeGuess**(), making sure that the guessedLetters list stores its letters in alphabetical order. Document your code. *Hint*: use String.charAt() to access a character at a given position in a string. (8)

(40 marks; 40 minutes)

Question 4

a) Write a Java program for a phone book, which stores an *ArrayList* of phone information. Each phone information object stores two pieces of data – the name of the person with that phone and their phone number. (20)
Hint: you should implement <u>two</u> classes: one for the phone book (e.g. a `PhoneBook` class), and one for phone information (e.g. a `PhoneInfo` class).

b) Write a main() method, clearly showing how you can add *and* remove phone information from a phone book. (10)

c) Could the phone book be implemented using a different data structure other than an ArrayList?

Explain your answer in words, with diagrams and small code fragments. (10)

--- End of Examination ---