**The School of Computer Science & Software Engineering**
**CITS 1220 Software Engineering**

# *Sample Answers for 2009 Sample Mid-semester Test*

This test had 5 pages (including cover) and 6 questions.
Each question was worth 5 marks, for a total of 30 marks
This test was worth a maximum of ten percent (10%) of your final mark in CITS1220.

## QUESTION 1 [5 marks]

Complete Javadoc documentation for the following Java method from a `MobilePhone` class:

```
/**
 * Increment the credit attribute of a mobile phone by amount in $.
 * @param amount integer value of $ to add to account
 * unchecked IllegalArgumentException is thrown
 * if amount is zero or negative
 */
public void topUp(int amount) {
     if (amount <= 0) {
          throw new IllegalArgumentException(
                    "topUp amount must be greater than 0 cents");
     } else {
          credit = credit + amount;
     }
}
```

## QUESTION 2 [5 marks]

Sketch Java code to return the longest string from a given array of strings. For example, given `String[] name={"which", "word", "is", "the", "longest", "one"}` then `longest(name)` returns `"longest"` (since that has 7 characters).

```
public String longest(String[] names) {
     int max =0;
     for(int i = 1; i<names.length; i++){
          if(names[i].length()>names[max].length())
               max = i;
     }
     return names[max];
  }
```

**QUESTION 3** **[5 marks]**

Consider the following fragment of a Java program.

```
Customer c;
if (Math.random()<0.5)
     c = new CreditCardCustomer();
else
     c = new CashCustomer();
c.bill(34);
```

a) How are the classes `Customer, CreditCardCustomer` and `CashCustomer` related?

*Customer is the superclass of CredtCardCustomer and CashCustomer (or it may be an interface implemented by the other two classes).*

b) Assuming that `CreditCardCustomer` and `CashCustomer` have different `bill()` methods, explain how the Java interpreter chooses which method body to execute.

*The java virtual machine uses dynamic binding. This means that at runtime, when c.bill() is called, the java virtual machine examines the object referenced by c, and uses the corresponding implementation of the bill method.*

**QUESTION 4** **[5 marks]**

Consider a simple Java Swing Graphical User Interface (GUI) that consists of a label with "Please enter name" on it, a text field which is empty and a button with "submit" written on it. A user is expected to type in the text field (possibly using backspace to correct mistakes) and then press the submit button. Some of the Java code to produce the GUI and its behaviour is written by the GUI programmer but much comes from the Java Swing library classes. Explain briefly what the programmer has to write. *Note that you are not required to sketch the code, just to explain what is required.*

*The programmer must*

- *initialize a JFrame*
- *initialize a JLabel with "Please enter name"*
- *initialize a JButton, "submit"*
- *initialize a JTextField*
- *add the JLabel, JButton, and JTextField to the JFrame*
- *Implement the ActionListener interface, and write code for the actions to perform when the button is pressed*
- *Add the ActionListener to the JButton.*

**QUESTION 5** **[5 marks]**

```
public boolean inorder( int day1, int month1, int day2, int month2 )
throws IllegalArgumentException
```

This Java method `inorder` accepts a date given by day1 and month1, and compares that date with a second date given by day2, month2. The return value is true if day1 of month1 is earlier than day2 of month2 and false otherwise. An exception is thrown if either date is illegal (eg day1=30, month1=2).

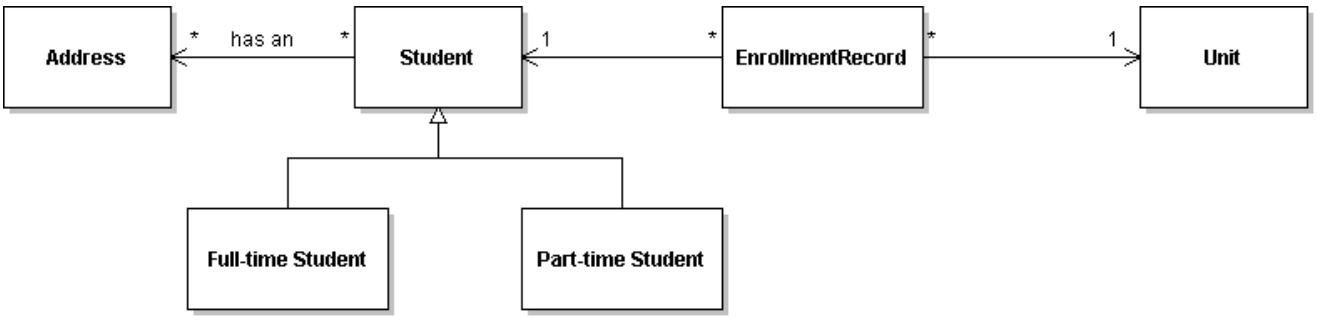
Give a set of 5 to 8 test cases for the method `inorder`. Show the inputs and expected outputs for each test case. For each test, state whether it tests normal, boundary or exceptional behaviour.

*An acceptable set of tests is*

| ***Tests*** | ***day1, month1, day2, month2*** | ***result*** | ***Type*** |
|---|---|---|---|
| *1* | *3, 5, 15, 5* | *true* | *Normal, same month* |
| *2* | *15, 5, 2, 6* | *true* | *Normal, later month* |
| *3* | *3, 6, 15, 5* | *false* | *Normal, wrong order* |
| *4* | *2, 1, 2, 1* | *false* | *Boundary, same day* |
| *5* | *30, 4, 1, 5* | *true* | *Boundary, month change* |
| *6* | *31, 4, 1, 5* | *Exception* | *Exceptional, illegal date* |
| *7* | *1, 15, 2, 15* | *Exception* | *Exceptional, illegal month* |

**QUESTION 6** **[5 marks]**

Summarise the information given by the UML class diagram below, using one or two sentences for each association between classes.




- *A student has 0 or more addresses*
- *More than one student may be registered at an address*
- *A Full-time Student **is a** student*
- *A Part-time Student **is a** student*
- *Each EnrollmentRecord is related to one student and one unit*
- *Each Student appears in 0 or more EnrollmentRecords*
- *Each Unit appears in 0 or more EnrollmentRecords*
- *Each student is enrolled in 0 or more units and each unit has 0 or more students enrolled.*