

**THE OPEN UNIVERSITY OF SRI LANKA
DEPARTMENT OF ELECTRICAL & COMPUTER
ENGINEERING**



ECI4262 – OBJECT ORIENTED DESIGN AND PROGRAMMING

FINAL EXAMINATION - 2010/ 11

CLOSED BOOK

Date: April 02, 2011

Time: 1400 – 1700 hrs

<INSTRUCTIONS >

There are two sections; **SECTION A** and **SECTION B**.
SECTION A has 1 general question; **Q1** is mandatory.
SECTION B has 5 questions; **Q2** to **Q6**. Answer to 3 selected questions on your preference.

SECTION A

[Q1] (Compulsory question)

- (i) State the Object Oriented Programming(OOP) concepts and the uses of the same? *[04 marks]*
- (ii) Programming to an interface is a practice that is followed in OOP.
(a) State two advantages that is achieved with this practice.
(b) Explain with Java or C interfaces and classes using code snippets how to achieve this?
(c) Why is decoupling important in OOP? *[09 marks]*
- (iii) Write a code snippet to demonstrate singleton design pattern with Java or C. *[08 marks]*
- (iv) Draw the class diagram for the (ii)(b.) code snippets *[08 marks]*
- (v) Write the output for the following code snippet. *[05 marks]*
- ```
int c = 1;
 for (int i = 1; i <= 5; i++) {
 for (int j = i; j < 5; j++) {
 System.out.print(" ");
 }
 for (int k = 1; k <= c; k++) {
 if (k % 2 == 0)
 System.out.print(" ");
 else
 System.out.print("#");
 }
 System.out.println();
 c += 2;
 }
```

**SECTION B**

Select and answer to any of 3 questions from Q2 – Q6

[Q2]

- (i) State the types (categories) of design patterns each with example. [02 marks]
- (ii) Explain the iterator pattern using a java code snippet (any loop with a collection) [05 marks]
- (iii) What are the common and different characteristics of the Bridge and the Strategy design patterns? [05 marks]
- (iv) When should the delegation pattern used over inheritance to extend a class's behavior? [05 marks]
- (v) Explain an Object Relational Mapping frame work and its uses as you see. [05 marks]

[Q3]

Consider the following code segment to answer to the question (i) and (ii).

```

/** Bank Interface */
public interface Bank {
 public abstract String getAccount();
}
/** Commercial Bank class */
public class CommercialBank implements Bank {
 public String getAccount() {
 return "Commercial Savings Account";
 }
}
/** Seylan Bank class */
public class SeylanBank implements Bank {
 public String getAccount() {
 return "Seylan Savings Account";
 }
}
/** Customer class */
public class Customer {
 private Bank bank;
 public Investor(Bank bank) {
 this.bank = bank;
 }
 public void setBank(Bank bank) {
 this.bank = bank;
 }
 public String openAccount() {
 return "I have an account of " + this.bank.getAccount();
 }
}

```

- (i) Draw the class diagrams for the above class structure and identify the relationships among those. [08 marks]
- (ii) What are the key components in Use Case diagrams? Explain using a suitable scenario (With drawing the use case diagrams for the given scenario). [08 marks]
- (iii) Differentiate between the following;  
 a.) 'class' and 'object'  
 b.) 'generalization' and 'association' relationship [06 marks]

[Q4]

- (i) Write the Java code segment to reverse the elements of an array. [07 marks]
- (ii) Write the Java code to calculate the average grade for 4 subjects using a loop. Also prompt the user for the marks for each subject and display the final average. [12 marks]
- (iii) Write the output of the below code snippet. [03 marks]

```

int n = 2;
int i = 1;
System.out.println("The table of " + n);
System.out.println("=====");
while(i <= 10) {
 int t = n * i;
 System.out.println(n + "*" + i + " = " + t);
 i++;
}

```

[Q5]

- (i) Briefly outline the purpose of using access specifiers to designate the members of a class as private, protected or public. [05 marks]
- (ii) Distinguish between single and multiple inheritances. Use suitable diagrammatic examples to illustrate your answer. [08 marks]
- (iii) What is the purpose of defining a method as abstract? [05 marks]
- (iv) What is meant by the term signature in describing a method? [04 marks]

[Q6]

(i) Explain the meaning of the following terms within the context of an object oriented programming language:

- a) method      b) message

[06 marks]

(ii) What advantages do object-oriented languages gain through the implementation of the above (in Q6 (i)) features?

[04 marks]

(iii) Explain how overloading and overriding contribute to the implementation of polymorphism in object oriented languages.

[06 marks]

(iv) State the most suitable UML diagram to fulfill the below requirements.

- a.) to show the physical relationship between software components and the hardware of a system delivered
- b.) to plan project activities such as developing new functionalities or test cases

[06 marks]