INTI
INTERNATIONAL COLLEGE PENANG (507232-U)
LAUREATE INTERNATIONAL UNIVERSITIES

FINAL Examination Paper

(COVER PAGE)

Session	:	**Apr 2013**
Programme	:	DIPLOMA IN ELECTRICAL AND ELECTRONICS ENGINEERING
Course	:	**CSC 2181: OBJECT ORIENTED PROGRAMMING IN JAVA**
Date of Examination	:	31 July 2013
Time	:	8a.m. – 10a.m. Reading Time : Nil
Duration	:	2 Hours
Special Instructions	:	

This paper consists of **SIX (6)** questions. Answer any **FOUR (4)** questions in the answer booklet provided. All questions carry equal marks.

Materials permitted	:	Nil
Materials provided	:	Answer Booklet
Examiner(s)	:	**Lim Chai Kim**
Moderator	:	Dr. Sukumar Letchmunan

This paper consists of _9__ printed pages, including the cover page.

INTI INTERNATIONAL COLLEGE PENANG

DIPLOMA IN ELECTRICAL AND ELECTRONIC ENGINEERING

CSC2181: OBJECT-ORIENTED PROGRAMMING IN JAVA

FINAL EXAMINATION: APR 2013 SESSION

Instructions: This paper consists of **SIX (6)** questions. Answer any **FOUR (4)** questions in the answer booklet provided. All questions carry equal marks.

Question 1

(a) Write a snippet of code to display the following series using a **nested for loop**.

```
0 1 2 3 4
0 1 2 3
0 1 2
0 1
0
```

(5 marks)

(b) Trace the output of the following code:

```
int fb1 = 0, fb2 = 1, sum = 0;
System.out.print(fb1 + " " + fb2 + " ");
do {
    sum = fb1 + fb2;
    fb1 = fb2;
    fb2 = sum;
    System.out.print(sum + " ");
} while (sum < 34);
```

(5 marks)

(c)

```
1   public TestQ {
2
3       public static double calculateY(double a, double b){
4           c = (a * b) / 2;
5       }
6
7       public static void main(String args) {
8
9           int y = calculateY(2, 3);
10          system.out.println(y);
11      }
12
13  }
```

Identify **FIVE (5)** syntax errors in the code above. Rewrite the code with correction.

(5 marks)

(d) Rewrite the following code using a "**switch**" flow structure:

```
if(type==1)
{
    System.out.println("Type A");
}
else if(type==2)
{
    System.out.println("Type B");
}
else if(type==3)
{
    System.out.println("Type O");
}
else if(type==4)
{
    System.out.println("Type AB");
}
else
{
    System.out.println("Unknown type");
}
```

(5 marks)

(e) Explain **ONE(1)** difference between a class and an object. Show example code to support your explanation.

(5 marks)

Question 2

(a) Write a full Java program to get the user's name, and then greet the user by name. Use JOptionPane to display the input dialog box and a separate dialog box to greet the user.

(5 marks)

(b) Write code to illustrate an example of **method overriding**.

(5 marks)

(c) Write code to illustrate an example of **method overloading**.

(5 marks)

(d) Describe **TWO (2)** differences between "**abstract class**" and "**interface**".

(5 marks)

(e) List **TWO (2)** possible ways to initialize a 2 dimensional integer array of size [2][3]. Provide an example code for this.

(5 marks)

Question 3

(a) The following declaration is found in a main method:

```
Frog a = new Frog( );
Frog b = new Frog ("Kermit", "Green",);
Frog c = new Frog ("Keroppi", "Blue", "Japan");
```

Write a declaration of class Frog with private attributes and public constructors.

(6 marks)

(b) State **ONE (1)** benefit of defining mutator methods in a class. Illustrate the benefit with an example mutator code using the class definition given in question 3(a) above.

(7 marks)

(c) Write a full Java program to calculate the sum of integers. Using java.util.Scanner, get the user's input of integers. The user can input any amount of integers. To stop, the user keys in "Quit". Here is an example output of the program:

```
Integer summation program.
Enter as many integers as desired.
To stop input of integers and display the sum, type "Quit".
Enter integer:
1
Enter integer:
2
Enter integer:
3
Enter integer:
Quit
Sum: 6
```

(8 marks)

(d) List **FOUR (4)** names that CANNOT be used as an identifier (variable or method name). Briefly state the reason it is invalid.

(4 marks)

Question 4

(a)(i) Write a class **RandomGen**, with a public static method **generateRandomNum(int min, int max)** that will return a random number between min and max (inclusive).

(5 marks)

(a)(ii) Test the method created in question 4(a)(i) in the main method of the same class. Simulate the rolling of a die and the throw of a coin. You do not need to write the import statements, just the main method. Here is a sample output of the program:

```
Result of rolling a die (1 to 6): 4
Result of throwing a coin (Head/Tail): Head
```

(5 marks)

(a)(iii) Test the method created in question 4(a)(i) in the main method of a different class. Ask the user to guess the result of a coin throw and display the result of whether the guess is correct or not. You do not need to write the import statements, just the main method. Here is a sample output of the program:

```
What is your guess for a coin throw? Head or Tail?
Head.
Result: Head. You are right.
```

If the user types in an invalid guess, display an error message and end the program. User input checking is case-insensitive.

(6 marks)

(b)
```
public interface Animal{

    public void makeNoise();
    public boolean isCarnivore();

}
```

Define a class, **Cat** that implements the above interface.

(5 marks)

(c) List **FOUR (4)** member level access modifiers.

(2 marks)

(d) List **FOUR(4)** checked exceptions in Java.

(2 marks)

Question 5

(a)(i) Explain in your own words the input, output and process of the following method:

```
public static boolean mysteryMethod(long n) {
    if (n == 1) {
        return false;
    }
    if (n == 2) {
        return true;
    }
    if (n % 2 == 0) {
        return false;
    }
    for (int i = 3; i <= Math.sqrt(n); i++) {
        if (n % i == 0) {
            return false;
        }
    }
    return true;
}
```

(4 marks)

(a)(ii) Below is a main method calling the method `mysteryMethod()` given in question 5(a)(i) above. Trace the output of the following method:

```
public static void main(String[] args) {
      System.out.println(mysteryMethod(10));
      System.out.println(mysteryMethod(7));
      System.out.println(mysteryMethod(165437656));
}
```

(3 marks)

(b)(i) Trace the output of the following main method:

```
public class Xxx {

   public String field1;

   public Xxx(String field1){
      this.field1 = field1;
   }

   public String print()  {
      System.out.println("field1: "+field1);
   }
}

public class Yyy extends Xxx{

   public String field2;

   public Yyy(String field1, String field2){
      super(field1);
      this.field2 = field2;
   }
   public String print()  {
      super.print();
      System.out.println("field2: "+field2);
  }

  public static void main(String[] args)
  {
      Xxx x, x1;
      x = new Xxx("abc");
      x1 = new Yyy("pqr", "stu");

      x.print();
      x1.print();
  }
}
```

(3 marks)

(b)(ii) State the Object-Oriented Programming concept that the code in question 5(b)(i) above exhibit. Briefly explain in your own words **ONE (1)** benefit of such a feature in Java.

(3 marks)

(c) Define the term "**package**" and explain how it is useful.

(5 marks)

(d)

```
1  public class SuperClass {
2    private String field1;
3    public String field2;
4
5    public SuperClass(String field1, String field2)
6    {
7       this.field1 = field1;
8       this.field2 = field2;
9    }
10 }
11
12 public class SubClass extends SuperClass{
13 {
14    public String field3;
15
16    public SubClass(String f1, String f2, String f3)
17    {
18       field1 = f1;
19       field2 = f2;
20       field3 = f3;
21    }
22 }
```

Find **ONE (1)** syntax error in the code above. Explain the error and suggest **TWO (2)** ways the error can be corrected.

(7 marks)

Question 6

(a)(i)
```
public class Book {
   private String title;
   private int yearOfPublish;

   public Book(String title, int yearOfPublish) {
      this.title = title;
      this.yearOfPublish = yearOfPublish;
   }
   public void setTitle(String title ) {
      this.title = title;
   }
   public int getYearOfPublish( ) {
      return yearOfPublish;
   }
   public String getTitle( ) {
      return title;
   }
}
```

Assume you have a class, **Library**, where an array of Book is declared and already instantiated and initialized as follows:

```
Book libraryBook[ ] = new Book[100];
```

Now complete the method removeOldBook below which will remove the title of the book and replace it with "Empty", if the yearOfPublish is less than 1945.

```
public void removeOldBook(Book[] libraryBook){
}
```

(5 marks)

(a)(ii) Based on the result of Question 6(a)(i) above, calculate the amount of "Empty" spaces so you can order new books. Write a method calculateEmptySpace which will return the number of empty spaces.

```
public int calculateEmptySpace(Book[] libraryBook){
}
```

(7 marks)

(b) What is a NumberFormatException? Write code to show the use of NumberFormatException.

(5 marks)

(c) Trace the output of the following code:

```
int j=3;
System.out.println(++j);
System.out.println(j++);
System.out.println(j);
j*=2;
System.out.println(j);
j = j + 6 / 3;
System.out.println(j);
float d = (float) j * 2 / 5;
System.out.println(d);
```

(3 marks)

(d) List **FOUR (4)** categories of operators in Java. For each category give **THREE (3)** example operators.

(5 marks)

--THE END—

CSC2181/Apr 2013/Lim Chai Kim