INTI International College Penang

FINAL
ALTERNATIVE ASSESSMENT

(COVER PAGE)

Session	:	**August 2020**
Programme	:	Diploma in Electrical & Electronic Engineering (DEEI)
Course	:	**CSC2181: Object-Oriented Programming in Java**
Date of Examination	:	12 December 2020 (Saturday)
Time	:	8.00am – 11.00am Reading Time : Nil
Duration	:	3 Hours

Special Instructions :

This paper consists of **FOUR (4)** questions. Answer **ALL** questions in the answer booklet provided.

Material permitted :
Non-Programmable Scientific Calculator

Materials provided :
Nil

Examiner(s)	:	**Christine Retnabai**
Chief Moderator	:	Nadhrah Binti Hadi

*This paper consists of **8** printed pages, including the cover page*

INTI INTERNATIONAL COLLEGE PENANG

DIPLOMA IN ELECTRICAL AND ELECTRONICS ENGINEERING PROGRAMME (DEEI)
CSC2181: OBJECT ORIENTED PROGRAMMING IN JAVA
FINAL ALTERNATIVE ASSESSMENT: AUGUST 2020 SESSION

Instructions: This section consists of **FOUR (4)** questions. Answer **ALL** questions in the answer booklet.

Question 1

a) Write a For loop that will prompt a user to input a positive integer. It should then print the multiplication of the table of that number.

(6 marks)

b) What will be the output of the following program?

```
public static void main(String[] args) {
 int i= 1;
 int j = 20;
 int k = 3;

 while (i<j) {
 k+= (i*j);
 i=i*2;
 j--;
 }

 System.out.println ("i = " + i + "j = " + j + "k = " + k);

  }

   }
```

(6 marks)

c) Write an example code with a parent class and a child class to illustrate method overriding.

(5 marks)

d) Assume we have three classes: Person, Teacher, and Student. Teacher and Student are both subclasses of Person. Which of the following assignments are legal, and why?

```
t = new Teacher ( );
p = t;
s = (Student) t;
s = (Student) p;
p = new Student ( );
t = new Person ( );
t = p;
```

(6 marks)

Question 2

a) What is an abstract class? Write a JAVA program where there is a Carnivor class that makes a roar sound using abstract class and explain how abstract class works.

(5 marks)

b) Write a JAVA program that prints the Fibonacci series up to a user input number.

Sample input and output of the program:

Enter the number of Fibonacci series you require: 7
Fibonacci series up to 7 numbers: 0 1 1 2 3 5

(10 marks)

c) Write a class, call it GradesCount, to read a list of grades from the keyboard (integer numbers in the range 0 to 100). Prompt the user with "Please enter a grade between 0 to 100 or -1 to quit: " each time before reading the next integer. Store each grade in a A, B, C, D or F Vector as follows: 90 to 100 = A, 80 to 89 = B, 70 to 79 = C, 60 to 69 = D, and 0 to 59 = F. (Hint: You cannot store ints as Vector elements, but you can store Integers.) Output the total number of grades entered, the number of A, B, C, D and F, and a list of the A's. For example, if the input is as below:

38
86
92
55
83
42
90
-1

then the output should be:

Total number of grades = 7
Number of A = 2
Number of B = 2
Number of C = 0
Number of D = 0
Number of F = 3
The A grades are: 92, 90

(15 marks)

Question 3

a) Without using an IF Statement to code, write a function to find the largest value among **TWO (2)** numbers. Include a main program to demonstrate the function.

(5 marks)

b) Provide the output of the following codes:

i)
```
class Superme {
public int index = 1;
}
Class App extends Superme{

public App(int index){
index = index;
}
}

public static void main (String[] args){
App ObjApp = new App(5);
System.out.println(ObjApp.index);

}
}
```

ii)
```
class TestIt {
protected int x, y;

}
class Main {
public static void(String args[]){
TestIt app = new TestIt();
System.out.println(app.x + " " + app.y)
}
```

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

for(int index = 0; true; index++) {
System.out.println("Welcome");
break;
}
}
}
```

iv)
```
class Output {
int i[] = {0};

public static void main (String args[]){
int i[] = {1};
alter(i);
System.out.println(i[0]);
                    }

public static void alter(int i[]){

int j[] = {2};
i=j;
}
}
```

v)
```
public void test (boolean a, boolean b) {

if (a) {
          System.out.println ("A");
          } else if (a && b) {
          System.out.println ("A && B");
          } else {
          if (!b) {
          System.out.println ("!B");
          } else {
          System.out.println ("None");
          }
          }
          }
```

(10 marks)

Question 4

a) **Definition of Armstrong:**
The sum of cubes of each digit is equal to the number itself.

Create a program that is able to check whether the input number is Armstrong or not.

(8 marks)

b) What is wrong with the following class definitions? There are **THREE (3)** errors. Fix the error and provide explanation.

```
class Employee {
private String name;
private int id;
public Employee (String name, int id) {
this.name = name; this.id = id;
}
public String getName ( ) {
return name;
}
public int getId ( )
{
return id;
}
}

class Manager extends Employee {
private String department;
     public Manager (String name, int id, String department)
     {
     this.name = name;
     this.id = id;
     this.department = department;
     }
 public void getDepartment ( ) {
     return department;
     }

     }
```

(5 marks)

c) Explain on what this piece of code can do and provide the output.

```
Random rand = new Random ( );
String table = "";
String row = "";
for (int i = 1; i < 4; i++) {
          for (int j = 1; j < 4; j++) {
               int number = rand.nextInt(50));
               row += number + " ";
               }
               table += row + "\n"; row = "";
               }
               System.out.println(table);
```

(4 marks)

d) Explain what has gone wrong with the following code and provide a more accurate code.

```
public interface Figures {
public void printMessage (String s) {
System.out.println ("This figure is " + s );
}
public double area ( ) ;     // to calculate the area of the figure
}
```

(3 marks)

e) Name **THREE (3)** important characteristics of a controller.

(3 marks)

f) In computer science, a Stack is a LIFO (last in, first out) data structure. Objects most recently inserted into a stack are the first objects removed from the stack. One way to design a stack data structure is to build it as a chain of node objects (like a linked list). The first part of this question involves designing an appropriate node class. Your node class should take advantage of parameterized types such that any type of object can be placed inside the node. Minimally, you should have appropriate class variables, constructors, and a set of accessors/mutators for each of your class variables. Fill in the Node class node.

(9 marks)

~THE END~