RMIT University
School of Computer Science and Information Technology

Final Examination **2005 Semester 2**

COSC1095 Programming Principles 2J
COSC1295 Java for Programmers

DATE: 2nd November 2005

Writing Time Allowed: 2 Hours

Reading Time: 5:45pm – 6:00pm

Writing Time: 6:00pm – 8:00pm

Number of Page: 11 pages (include this page)

Instructions to Candidates:

- Read these instructions carefully.
- Answer **all** questions. All answers should be written on exam answer sheets.
- This is a **closed book** exam. You shall not use published textbooks, class notes, copies of assignment work or any other written or printed references.
- You **cannot** use any electronic devices – no calculators, palmtop computers, nothing electronic. Mobile phones **must** be switched **off**.
- This paper accounts for **60%** of your final assessment in the course and will be marked out of 120. As an approximate guide a mark equals to a minute of your time.
- Read the questions carefully and try to clarify any ambiguities during reading time. If you need to make any assumptions then state them in your answer.

Part A: Multiple Choices (30 marks: 3 marks per question)

- Please write the answer on your answer sheet.
- **One or more correct choices are possible**. In such cases, select **all** the possible answers.

1. Suppose there is a class called **Company**, which of the following statement(s) about its constructor is/are incorrect?

A. To define a constructor:
public void Company(String companyName){ / some code…*/ }*

B. To define another constructor:
public Company(int companyRegNumber){ / some code…*/ }*

C. To use a constructor:
Company.Company("IBM");

D. To use a constructor in its subclass:
super(regNumber);

2. Which of the statement(s) in the ***main*** method below will cause compilation error(s)?

```
class NumberOne {
   public void print(){System.out.println("This is  NumberOne");}
}

abstract class NumberTwo {
   public void print(){System.out.println("This is NumberTwo");}
}

class NumberThree  extends NumberTwo{
   public void print(){System.out.println("This is NumberThree");}
}

abstract class NumberFour extends NumberTwo {
   public void print(){System.out.println("This is  NumberFour");}
}

//Continued …
```

```
public class TestInterface{
  public static void main (String[] args) {
       Object  p;                    // statement 1
       p = new NumberOne();          // statement 2
       p = new NumberTwo();          // statement 3
       p = new NumberThree();        // statement 4
       p = new NumberFour();         // statement 5
  }
}
```

A. statement 1

B. statement 2

C. statement 3

D. statement 4

E. statement 5

F. None of the above

3. Consider the following code. Which methods will NOT be executed, if method2() caused a NoSuchMethodException.

```
try
  {
     method1();
     method2();
     method3();
  }
 catch (MyOwnException  e1)
 {
     method4();
 }
 catch (Exception e2)
 {
      method5();
 }
 finally
 {
      method6();
 }
 method7();
```

A. method3()

B. method4()

C. method5()

D. method6()

E. method7()

4. In the program below class **A** has one method *method1()*. The subclass **B** has another two methods *method2()* and *method3()*. Which of the statement(s) below in the *main* method of the **Test** class will result compilation errors?

```
public class Test  {
   public static void main(String[] args) {
        A myObject = new B();

        myObject.method1();            // statement 1

        myObject.method2();            // statement 2

       (B) myObject.method2();         // statement 3

       ((B) myObject).method3();       // statement 4
   }
}
class A {
   public void method1() {    }
}
class B extends A{
   public void method1() {    }
   public void method2() {    }
   public void method3() {    }
}
```

A. statement 1

B. statement 2

C. statement 3

D. statement 4

E. None of the above

5. In the Java Collection Framework, which collection(s) cannot contain duplicated elements?

A. Set

B. Vector

C. List

D. LinkedList

E. SortedSet

6. Which of the following statement(s) about **Event Driven Programming** is/are true?

A. Events in Java are objects

B. One event source might generate different kinds of events.

C. One event listener could listen to multiple event sources.

D. An event listener needs to register itself with the event source.

E. All Above

7. Which of the statement(s) in Class B is/are **invalid**?

```
class A {
  private int a;
  protected int b;
  public int c;
}
.......................
class B extends A { // in the same package, B has no extra instance variables
 void increment() {
    this.a++;           //statement 1
    this.b++;           //statement 2
    c++;                //statement 3
    this.c++;           //statement 4
    this.d++;           //statement 5
 }
}
```

A. Statement 1

B. Statement 2

C. Statement 3

D. Statement 4

E. Statement 5

8. In the following code, which method(s) below must be implemented in class **D** to make it a concrete class?

```
abstract class A {
    public abstract void methodOne();
    public void methodTwo(){}
}

interface B {
  public void methodThree();
}

interface C extends B {
   public void methodFour();
   public void methodFive();
}

class D extends A implements C {
……
}
```

A. methodOne()

B. methodTwo()

C. methodThree()

D. methodFour()

E. methodFive()

9. To achieve multithreading in a Java GUI application, one could:

A. “implements” the Thread interface

B. “implements” the Runnable interface

C. “extends” the Thread class

D. “extends” the Runnable class

E. none of the above

10. What is the output of the following code?

```
class Parent {
    public void myPrint(int i)
      { System.out.print( i +" is printed " ); }

    public void myPrint(int i, String name)
      { System.out.print( i +" is printed for "+ name +"."); }
}

class Child extends Parent{
    public void myPrint(int i)
      { super.myPrint(i); }

    public void myPrint(int i, String name)
      { this.myPrint(i);
        System.out.print( "as requested by " + name +".");
      }
}

public class TestMethod{
   public static void main (String[] args) {
       Parent  p = new Parent();
       p.myPrint(10, "Sam");
       p = new Child();
       p.myPrint(20, "Bill");
   }
}
```

A. 10 is printed for Sam. 20 is printed as requested by Bill.
B. 10 is printed for Sam. 20 is printed for Bill.
C. 10 is printed as requested by Sam. 20 is printed as requested by Bill.
D. - Compilation Errors -
E. - Runtime Errors -

Part B: Questions and Short Programming Tasks (40 marks)

1. What is a *Synchronized* Thread? What is a *deadlock*? Explain with an example.
.

(6 marks)

2. What is the output of the following program? Show your works.

(8 marks)

```
int[][] m = new int[4][4];
for (int j=0; j<4; j++)
     for (int i=0; i<4; i++)
             if (i >= j)
                     m[i][j] = 1;
             else
   m[i][j] = 0;

  for (int i=1; i<4; i++) {
    for (int j=0; j<3; j++)
            System.out.print("    "+m[i][j]);
            System.out.println();
  }
```

3. In a relational database you have the following table.

StudentID	**Courses**
S123456	Java
S654321	DataBase
S109431	Java

Table Name: **Enrolment**

The URL connection string is “jdbc:odbc:TESTDB”
User name: “one”, Password “sesame”
And the JDBC driver is at “sun.jdbc.odbc.JdbcOdbcDriver”

Briefly discuss how to create a Connection to the database server, and then query the table and put the information in a data structure so that the enrolment details can be listed. You can use code segments for the discussion.

(10 marks)

4. Implement a class **MyLinkedList** which can maintain a three-node linked list. Each node stores an integer. This class should support the following test program.

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

   MyLinkedList test =  new MyLinkedList(10, 20, 15);

   test.print();                    //output 10 20 15
   test.delete(10);                 //delete 10 from  the linked list
   test.print();                    //output 20 15

    }
}
```

(16 marks)

Part C: Extended Programming Tasks (50 marks)

An Academic Management System

The following is a simple academic administration system for a university. It demonstrates the use of inheritance and polymorphism in Java programming.

There are mainly 6 classes in this system. These classes are **Program**, **Postgraduate**, **Undergraduate**, **InternationalUndergraduate, InternationalPostgraduate** and **Test.**

Following is the design of the system with descriptions of each mentioned class. Your task is to write or to complete the classes / interfaces using the information given.

(a) **Program** class **(10 marks)**

This is an abstract class. It does not represent a particular academic program but rather it serves as the superclass of **Postgraduate** and **Undergraduate**. This class contains information that applies to all programs. Each program includes a program code (eg. "BP162"), number of enrollment (e.g 104 students), running cost per student (e.g. $3000 per head) and tuition fees (e.g. $6000 per student). These properties can be retrieved by the methods *getCode()*, *getEnrolNum(), getCostPerHead()* and *getTuitionFees()* of this class respectively. This class also contains an **abstract** method *computeCost ()* to calculate the total cost of running the program.

(b) **Undergraduate** class **(10 marks)**

This concrete class is inherited from the **Program** class. It represents an undergraduate program. It has a constructor that uses the constructor of its superclass. It has no more methods than its superclass has. For an undergraduate program which has less than 50 students enrolled, the government will subsidize $50K, which means the running cost of that program is reduced by $50,000. For a program of size in between 50 and 100, the government will support $70K. For a program which has more than 100 students, the government will fund $80K.

Note: no undergraduate program is allowed to have less than 10 students. You might need to define your own **EnrolmentTooLowException.**

(c) **Postgraduate** class **(10 marks)**

Also inherited from the **Program** class. It represents postgraduate programs. It uses the constructor of its superclass. It has no extra methods. The government does not support postgraduate programs. Instead, the university needs to pay $1000 per student as government tax for a postgraduate program if the tuition fees of that program is more than $8,000 per student. No minimum enrollment requirement for postgraduate programs.

(d) **InternationalUndergraduate, InternationalPostgraduate** classes **(10 marks)**

Both of them implement an interface **International.** They are programs for international students. For any international programs, the government will charge a fixed percentage of the tuition fees as tax. The current tax rate is 5%. Other than that, international programs have no difference with their local counterparts.

(e) **Test** class **(10 marks)**

This is a driver class to test the above classes/interfaces. Complete the follow code, so it could:

- loop through every program to report the total cost of running these programs
- report the total cost of international programs if there is any.

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

      //Code  needed here

      list[0] = new Undergraduate("BP162", 20, 5000, 9000);
      list[1] = new Postgraduate("MC062", 100, 5000, 10000);
      list[2] = new InternationalUndergraduate("BP152Intl", 50, 5000, 13000);
      list[3] = new InternationalPostgraduate("MC055Intl", 30, 6000, 12000);
      list[4] = new InternationalPostgraduate("GD105Intl", 5, 9000, 18000);

      //Code needed here

  }
}
```

========== The End ==========