



EXAMINATIONS — 2002

MID-YEAR

COMP 102
INTRODUCTION TO
COMPUTER PROGRAM
DESIGN

Time Allowed: 3 Hours

Instructions: Attempt ALL Questions.

Answer in the appropriate boxes if possible — if you write your answer elsewhere, make it clear where your answer can be found.

The exam will be marked out of 180.

Non-programmable calculators without full alphabet keys are permitted.

Non-electronic foreign language dictionaries are permitted.

Questions

	Marks
1. Understanding simple Java programs	[30]
2. Writing simple Java programs	[30]
3. Arrays	[30]
4. Files and recursion	[30]
5. Simple containers	[30]
6. Composition, containers, genericity	[30]

Question 1. Understanding simple Java programs

[30 marks]

(a) [3 marks] What will be printed when method AAA below is called?

```
public void AAA() {  
    int k = 6;  
    k = k*(k-1)/2;  
    System.out.print("Result = ");  
    System.out.println(k);  
}
```

(b) [3 marks] What will be printed when method BBB below is called with **4** as its argument?

```
public void BBB(int x) {  
    int y = 5;  
    y = x + y;  
    x = x - 1;  
    y = y * x;  
    System.out.println(y);  
}
```

(c) [4 marks] What will be printed when method CCC below is called with **12** as its argument?

```
public void CCC(int n) {  
    for (int k = 5; k < n; k = k + 2) {  
        System.out.print(k + " ");  
    }  
}
```

(d) [6 marks] Consider the following method definition:

```
public void DDD(int n) {  
    if ( n > 0 )  
        if ( n < 8 )  
            System.out.print("navy ");  
        else  
            if ( n > 20 )  
                System.out.print("dark ");  
            else  
                System.out.print("light ");  
    System.out.println("blue");  
}
```

(i) [2 marks] What will be printed when DDD is called with **6** as its argument?

(ii) [2 marks] What will be printed when DDD is called with **18** as its argument?

(iii) [2 marks] What will be printed when DDD is called with **0** as its argument?

(e) [4 marks] Consider the following method definition:

```
public void EEE(String s) {
    int k = 0;
    while ( true ) {
        char c = s.charAt(k);
        if ( c == '@' ) break;
        k = k+1;
    }
    System.out.print(k);
}
```

(i) [2 marks] What will be printed when EEE is called with "lindsay@mcs.vuw.ac.nz" as its argument?

(ii) [2 marks] What will happen when EEE is called with "http://www.mcs.vuw.ac.nz/" as its argument?

(f) [4 marks] What will be printed when method FFF below is called?

```
public void FFF() {
    for (int k = 0; k<8; k++) {
        if ( (k%2 == 0) || (k%3 == 0) )
            System.out.print(k + " ");
    }
}
```

(Remember that $x\%y$ is the remainder of dividing x by y .)

(g) [3 marks] Consider the following class definition:

```
class Counter1 {
    public Counter1() {
        Count();
        Count();
    }
    public void Count() {
        int k = 0;
        for (int i = 1; i<4; i = i+1) {
            System.out.print(k + " ");
            k = k+1;
        }
    }
}
```

What will be printed when the constructor Counter1 is called?

(h) [3 marks] Consider the following class definition, where Counter2 is the same as Counter1 above except for the declaration of k:

```
class Counter2 {
    private int k = 0;
    public Counter2() {
        Count();
        Count();
    }
    public void Count() {
        for (int i = 1; i<4; i = i+1) {
            System.out.print(k + " ");
            k = k+1;
        }
    }
}
```

What will be printed when the constructor Counter2 is called?

(e) [8 marks] The following program reads the height and width of two rectangular panels, and prints their combined width and the height of the higher panel. You are required to complete the methods `getHeight`, `getWidth` and `isHigher`.

```
import javax.swing.*;

public class TwoPanels {
    public static void main(String[] args) {
        Panel left = new Panel();
        Panel right = new Panel();
        int w = left.getWidth() + right.getWidth();
        String s = "Combined width is " + w + "\n";
        if ( left.isHigher(right) )
            s = s + "Height is " + left.getHeight();
        else
            s = s + "Height is " + right.getHeight();
        JOptionPane.showMessageDialog(null, s);
    }
}

class Panel {

    private int height;
    private int width;

    // Read the data for a new panel
    public Panel() {
        // The details don't matter for this question.
        // You are not required to implement this constructor.
    }

    // Return the height of this panel
    public int getHeight() {
        // See part (i).
    }

    // Return the width of this panel
    public int getWidth() {
        // See part (ii).
    }

    // Return true if this panel is higher than panel p, otherwise false
    public boolean isHigher(Panel p) {
        // See part (iii).
    }
}
```

(i) [2 marks] Complete the method `getHeight`.

```
// Return the height of this panel
public int getHeight() {

}
}
```

(ii) [2 marks] Complete the method `getWidth`.

```
// Return the width of this panel
public int getWidth() {

}
}
```

(iii) [4 marks] Complete the method `isHigher`.

```
// Return true if this panel is higher than panel p, otherwise false
public boolean isHigher(Panel p) {

}
}
```

Question 3. Arrays

[30 marks]

(a) [15 marks] Arrays of Integers

Consider the following program:

```
import javax.swing.JOptionPane;

public class ArrayEx{
    public static void main(String[] args){
        NumberArray n = new NumberArray();
        n.print();
        n.findMax();
    }
}

class NumberArray{
    private int[] nums = {4, 1, 2, 8, 5};

    public void print(){
        int i=0;
        System.out.println(nums[1]);
        System.out.println(nums[i]+nums[i+1]);
        System.out.println(nums[nums[0]]);
        System.out.println(nums[nums.length-1]);

        for (int j = 0; j <= (nums.length/2); j++){
            System.out.println(nums[j]);
        }
    }

    public void findMax(){
        //You need to write the code for this method in part (ii).
    }
}
```

(i) [7 marks] What will be printed when `n.print()` is called?

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

(b) [15 marks] Arrays of Objects

Consider the following program:

```
import javax.swing.JOptionPane;

public class ArrayObjectsEx{
    public static void main (String[] args){
        StudentOffice sOffice = new StudentOffice();
        sOffice.readRecords();
        sOffice.printStudents();
        sOffice.printGoodStudents();
    }
}

class StudentOffice{
    private Student[] studentsRecord;
    private int MAXStudents = 500;
    private int numStudents = 0;

    public StudentOffice(){
        studentsRecord = new Student[MAXStudents];
    }

    public void readRecords(){
        numStudents = 0;
        Student e = new Student();
        while ((e.enter()) && (numStudents < MAXStudents)){
            studentsRecord[numStudents] = e;
            numStudents++;
            e = new Student();
        }
        System.out.println("Number of students read: " + numStudents);
    }

    public void printStudents(){
        for (int i = 0; i < numStudents; i++){
            System.out.println(studentsRecord[i].getName());
        }
    }

    public void printGoodStudents(){
        // You need to write the code for this method in part (ii).
    }

    public double averageMark(){
        double totalmark = 0.0;

        for (int i = 0; i < numStudents; i++){
            double smark = studentsRecord[i].getMark();
            totalmark = totalmark + smark;
        }
        return totalmark/numStudents;
    }
}
```

```
class Student{
    private String name;
    private double mark;

    public String getName(){
        return name;
    }

    public double getMark(){
        return mark;
    }

    public boolean enter(){
        name = JOptionPane.showInputDialog("Enter name:");
        if (name == null) return false;
        String sMark = JOptionPane.showInputDialog("Enter mark:");
        mark = Double.parseDouble(sMark);
        return true;
    }

    public void print(){
        System.out.println("Name: " + name);
        System.out.println("Mark: " + mark);
    }
}
```


Question 4. Files and recursion

[30 marks]

(a) [16 marks] Reading from files

Suppose we have a file containing data about the daily foreign exchange rate between a New Zealand dollar and an Australian dollar. The data is saved in this format:

date/month/year:rate

For example, the file `data.txt` contains the following data for six days.

```
01/06/02:0.8409
02/06/02:0.8380
03/06/02:0.8300
04/06/02:0.8415
05/06/02:0.8275
06/06/02:0.8515
```

Note: You must answer the following two subquestions using the example file given above.

(i) [6 marks] What will the following program print out?

```
import java.io.*;

public class ReadFileEx1{
    public static void main(String[] args){
        try{
            FileReader inStream = new FileReader("data.txt");
            BufferedReader ins = new BufferedReader(inStream);

            for(int i=0; i<3; i++){
                String dataLine = ins.readLine();
                System.out.println(dataLine);
            }
            ins.close();
        }
        catch (IOException ex){
            System.out.println("i/o error: " + ex.getMessage());
            ex.printStackTrace();
        }
    }
}
```

(ii) [10 marks] What will the following program print out?

```
import java.io.*;

public class ReadFileEx2{
    public static void main(String[] args){
        try{
            FileReader inStream = new FileReader("data.txt");
            BufferedReader ins = new BufferedReader(inStream);
            int num=0;

            String dataLine = ins.readLine();

            while(dataLine!=null){
                num++;
                String s = dataLine.substring(9, dataLine.length());
                double n = Double.parseDouble(s);
                if (n<=0.84){
                    String ss = dataLine.substring(0, 8);
                    System.out.println(ss);
                }
                dataLine = ins.readLine();
            }
            ins.close();
            System.out.println(num);
        }
        catch (IOException ex) {
            System.out.println("i/o error: " + ex.getMessage());
            ex.printStackTrace();
        }
    }
}
```



(b) [6 marks] Writing to files

Complete the following program so that it creates a file called `myFile.txt` and saves the following information into the file.

```
*****HELLO WORLD!*****
```

```
import java.io.*;
public class WriteFileEx{
    public static void main(String[] args){
        try{
            // Create a file object for output

            //Write the data to the output file
            String data = "*****HELLO WORLD!*****";

            //Close the output file

        }
        catch (IOException ex){
            System.out.println("i/o error: " + ex.getMessage());
            ex.printStackTrace();
        }
    }
}
```

(c) [8 marks] Recursion

Consider the following program:


```
import javax.swing.JOptionPane;

public class RecursionEx{
    public static void main(String[] args){
        String s=JOptionPane.showInputDialog("Enter number:");
        int num = Integer.parseInt(s);

        int result = myMethod(num);
        System.out.println("myMethod(" +num +" ) is: "+ result);    //Print the result
    }

    public static int myMethod(int n){
        System.out.println(n);    //Print the argument each time myMethod is called
        if (n <= 2)
            return n;
        else{
            int f1= myMethod(n-2);
            int f2= myMethod(n-1);
            return f1 * f2;
        }
    }
}
```

Suppose that the user enters the number 5. What will this program print out?



Question 5. Simple containers

[30 marks]

Consider this Set class as shown in a lecture.

```
class Set {
    private String[] elements;
    private int maxsize = 100;
    public Set(int setsize) {
        maxsize = setsize;
        elements = new String[maxsize];
    }
    public Set() {
        elements = new String[maxsize];
    }
    public boolean Insert(String x) {
        if (Check(x))
            return true;
        int i;
        i = 0;
        while (i < maxsize) {
            if (elements[i] == null) {
                elements[i] = x;
                return true;
            }
            i++;
        }
        return false;
    }
    public boolean Check(String x) {
        int i;
        i = 0;
        while (i < maxsize) {
            if (elements[i] != null
                && elements[i].equals(x)) {
                return true;
            }
            i++;
        }
        return false;
    }
}
```

(a) [4 marks] What is printed by the program SetTester1 below?

```
class SetTester1 {
    public static void main(String[] args) {
        Set things = new Set();
        if (things.Insert("apple")) {
            if (things.Check("apple"))
                System.out.print("one ");
            if (things.Check("apple"))
                System.out.print("two ");
        }
        System.out.println("done");
    }
}
```

(b) [4 marks] What is printed by the program SetTester2 below?

```
class SetTester2 {
    public static void main(String[] args) {
        Set things = new Set(1);           // explicit parameter
        if (things.Insert("apple")) {
            if (things.Insert("orange"))
                System.out.print("yes ");
            else
                System.out.print("no ");
            if (things.Check("orange"))
                System.out.print("in ");
            else
                System.out.print("out ");
        }
        System.out.println("done");
    }
}
```

(c) [4 marks] What is printed by the program SetTester3 below?

```
class SetTester3 {
    public static void main(String[] args) {
        Set things = new Set();
        if (things.Insert(null))
            System.out.print("yes ");
        else
            System.out.print("no ");
        if (things.Insert(null))
            System.out.print("more ");
        else
            System.out.print("less ");
        if (things.Check(null))
            System.out.print("still ");
        else
            System.out.println("never");
    }
}
```

QUESTION 5 IS CONTINUED ON THE NEXT PAGE

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

Question 6. Composition, containers, genericity

[30 marks]

```

public class CarFinder {
    public static void main(String[] args) {
        CarMap fleet = new CarMap(); // Construct fleet
        while (true) {
            String cmd = JOptionPane.showInputDialog("Command: ");
            if (cmd == null) System.exit(0);
            if (cmd.equals("newcar")) {
                String r = JOptionPane.showInputDialog("Registration Num: ");
                Car c = new Car();
                if (!fleet.Insert(r, c))
                    System.out.println("Unable to insert!");
            }
            else if (cmd.equals("findcar")) {
                String r = JOptionPane.showInputDialog("Registration Num: ");
                Car c = fleet.Find(r); // Find in Fleet
                if (c != null)
                    c.PrintDetails();
                else
                    System.out.println("Not Found!");
            }
        }
    }
}

class CarMap {
    private String[] regs;
    private Car[] records;
    public CarMap() {
        regs = new String[5];
        records = new Car[5];
    }
    public boolean Insert(String reg, Car c) {
        if (Find(reg) != null) return false;
        for (int n = 0; n < 5; n++) {
            if (regs[n] == null) { // Insert compare
                regs[n] = reg;
                records[n] = c;
                return true;
            }
        }
        return false;
    }
    public Car Find(String reg) {
        // NOT SHOWN: see question 6 part e
        // should return car that matches the given registration number, or null
    }
}

class Car {
    private String make;
    private int year;
    public Car() {
        make = JOptionPane.showInputDialog("Make: ");
        String yearstr = JOptionPane.showInputDialog("Year: ");
        year = Integer.parseInt(yearstr);
    }
    public void PrintDetails() {
        System.out.println("Car: " + make + ", year is " + year);
    }
}

```


The rest of this question relates to the generic Map class below.

```

class Map {
    private Object[] keys;
    private Object[] values;
    private int maxsize;
    private int size = 0;
    private int pos;

    public Map(int setsize) {
        maxsize = setsize;
        keys = new Object[maxsize];
        values = new Object[maxsize];
    }

    public boolean Insert(Object k, Object v) {
        if (size == maxsize)
            return false;
        keys[size] = k;
        values[size] = v;
        size++;
        return true;
    }

    public Object Find (Object k) {
        // NOT SHOWN: see question 6 part h
        // should return value that matches the given key, or null
    }

    // Traversal methods
    public void getInit() {
        pos = -1;
    }
    public boolean getNext() {
        pos++;
        return (pos < size);
    }
    public Object getKey() {
        return keys[pos];
    }
    public Object getValue() {
        return values[pos];
    }
}

```

