



EXAMINATIONS — 2006

END-OF-YEAR

COMP 102

WITH SOLUTIONS
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 marks.

Non-programmable calculators without a full alphabetic key pad are permitted.

Non-electronic foreign language dictionaries are permitted.

(Documentation at the end of the paper has been removed from the answers)

(Spare pages have been removed from the answers)

Questions

	Marks
1. Understanding Java	[35]
2. Files	[12]
3. Programming with Loops	[15]
4. Implementing a Class	[15]
5. Arrays	[36]
6. Recursion	[30]
7. Inheritance	[37]

Question 1. Understanding Java

[35 marks]

(a) [3 marks] What will the following fragment of Java print out?

```

int me = 2;
int us = me + 5;
int them;
if ( us > 7 )
    them = us * 3;
else
    them = us - 8;
System.out.printf("me=%d, us=%d, them=%d\n", me, us, them);

```

```

| me=2, us=7, them=-1

```

(b) [3 marks] What value will be returned by the call basic("Hello")?

```

public int basic(String str){
    if ( str.equals("hello") )
        return str.length();
    if ( str.equalsIgnoreCase("hello") )
        return str.length() * 2;
    return 15;
}

```

```

| 10

```

(c) [3 marks] What will the following code fragment print to System.out?

Note, the first line constructs an array of length 9 containing the specified strings.

```

String[] values = {"ant", "bee", "cat", "dog", "eel", "fox", "gnu", "hen", "imp"};
for( int k=0; k<values.length; k++ ){
    System.out.print(values[k]+ " ");
}

```

```

| ant bee cat dog eel fox gnu hen imp

```

(Question 1 continued on next page)

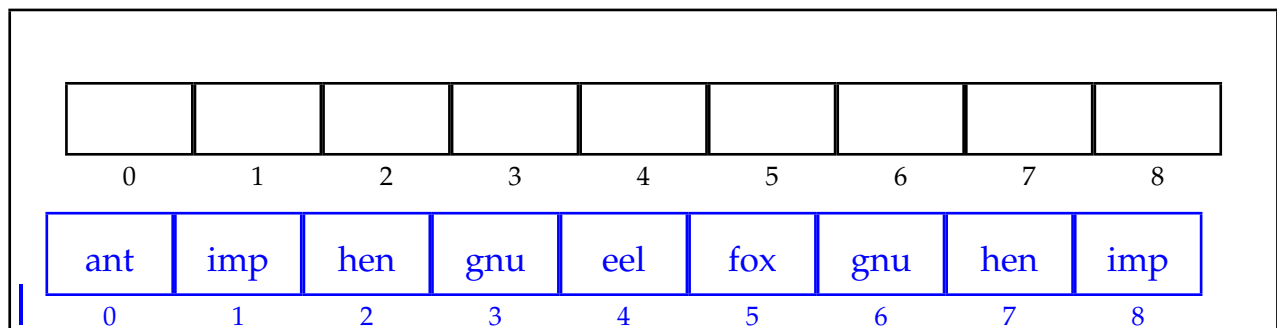
(Question 1 continued)**(d)** [5 marks] What will the following code fragment print to System.out?

```
String[] values = {"ant", "bee", "cat", "dog", "eel", "fox", "gnu", "hen", "imp"};
for ( int k=values.length-1; k>0; k=k/2 ){
    System.out.print(values[k]+ " ");
}
```

imp, eel, cat, bee

(e) [6 marks] Show the contents of the `values` array after the following code fragment has been run.

```
String[] values = {"ant", "bee", "cat", "dog", "eel", "fox", "gnu", "hen", "imp"};
for ( int k=1; k<values.length/2; k++){
    values[k] = values[values.length-k];
    values[values.length-k] = values[k];
}
```



(Question 1 continued on next page)

(Question 1 continued)

(f) [5 marks] Suppose the file "output.log" contains the following text:

```
Log of errors from exam.pas
Line 015: warning: function "error" not yet defined
Line 018: error: variable "size" not declared
Line 025: warning: invalid syntax for "+"
Line 029: error: function "error" has no arguments
Line 038: error: file not closed
Line 048: warning: program has implicit END.
Compile Failed
```

What will the following method print out if it is called?

```
public void readLog(){
    try{
        int errors=0;
        Scanner sc = new Scanner(new File("output.log"));
        while ( sc.hasNext() ){
            String line = sc.nextLine();
            if ( line.contains("error") ){
                errors++;
                System.out.println( line.substring(5,8));
            }
        }
        sc.close();
        System.out.println(errors + " Errors");
    }
    catch(Exception e){System.out.println("Reading log file failed");}
}
```

```
f e
015
018
029
038
5 Errors
```

(Question 1 continued on next page)

(Question 1 continued)

Consider the following Envelope class.

```
public class Envelope{
    // Fields
    private int size = 1;    // size must be 1, 2, or 3
    private int width;
    private double weight =0; //weight in grams
    private String address;

    // Constructor
    public Envelope(int sz){
        if (0 < sz && sz < 4)
            size = sz;
        System.out.println("New Envelope of size " + size);
    }

    // Methods
    public void setAddress(String s){
        if (address != null)
            System.out.println("Changing "+address);
        address = s;
    }

    public void addWeight(double w){
        weight = weight + w;
        if (weight > 100 * size){
            System.out.println("Envelope is now over-weight");
            address = address + "\n TOO HEAVY ";
        }
    }

    public String toString(){
        return ("Size "+ size + " envelope to "+address + " : "+weight +"gms");
    }
}
```

(Question 1 continued on next page)

(Question 1 continued)

(g) [10 marks] Given the *Envelope* class on the facing page, what will the following code fragment print out?

```
Envelope e = new Envelope(2);
e.setAddress("Lindsay, VUW");
e.addWeight(150);
System.out.println(e.toString ());

System.out.println ("-----");
e.setAddress("Pondy, SMSCS");
e.addWeight(80);
System.out.println(e.toString ());

System.out.println ("-----");
e = new Envelope(4);
e.setAddress("Marvin, SCPS");
e.addWeight(110);
System.out.println(e.toString ());
```

```
| New Envelope of size 2
| Size 2 envelope to Lindsay, VUW : 150.0gms
```

```
-----
| Changing Lindsay, VUW
| Envelope is now over-weight
| Size 2 envelope to Pondy, SMSCS
| TOO HEAVY : 230.0gms
```

```
-----
| New Envelope of size 1
| Envelope is now over-weight
| Size 1 envelope to Marvin, SCPS
| TOO HEAVY : 110.0gms
```

Question 2. Files

[12 marks]

Complete the following `countFile` method so that it reads the contents of a file and prints out the number of lines and the number of characters in the file (not counting the end of line characters). The argument to `countFile` is the name of the file to read.

```
public void countFile(String fname){
    try {

ANS:         int lines =0;
ANS:         int chars = 0;
ANS:         Scanner sc = new Scanner(new File(fname));
ANS:         while (sc.hasNext()){
ANS:             String line = sc.nextLine();
ANS:             lines++;
ANS:             chars = chars+line.length ();
ANS:         }
ANS:         sc.close ();

        System.out.printf("Lines: %d\nCharacters: %d\n", lines, chars);

    }
    catch(Exception e){System.out.println("File reading failed");}
}
```

Question 3. Programming with Loops

[15 marks]

(a) [5 marks] What will the following printTable method print out if it is called with an argument of 4?

```
public void printTable ( int n){
    for ( int k=0; k < n*n; k++){
        System.out.printf ("%2d ", k);
        if ( k==n )
            System.out.println ();
    }
}
```

0	1	2	3	4															
5	6	7	8	9	10	11	12	13	14	15									

(b) [10 marks] printTable was intended to print out an $n \times n$ table of the integers starting at 1. The argument specifies how many rows and columns in the table. For example, if the argument is 3, printTable should print out:

```
1 2 3
4 5 6
7 8 9
```

If the argument is 5, it should print out:

```
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
```

On the facing page, write a correct version of printTable so that it does what it is supposed to.

(Question 3 continued on next page)

(Question 3 continued)

```

public void printTable ( int n){

ANS:   public void printTable1 ( int n){
ANS:   for ( int i=1; i <= n*n; i++){
ANS:       System.out.printf ( "%2d ", i);
ANS:       if ( i%n == 0){
ANS:           System.out.println( );
ANS:   }}}
ANS:
ANS:   public void printTable2 ( int n){
ANS:       for ( int row=0; row < n; row++){
ANS:           for ( int col=0; col < n; col++)
ANS:               System.out.printf ( "%2d ", row*n+col+1);
ANS:           System.out.println( );
ANS:       }}
ANS:
ANS:   public void printTable4 ( int n){
} ANS:   int i = 1;
ANS:   int col = 1;
ANS:   while ( i<=n*n){
ANS:       System.out.printf ( "%2d ", i++);
ANS:       if ( col++ == n){
ANS:           System.out.println( );
ANS:           col=1;
ANS:       }}}
ANS:
ANS:   public void printTable3 ( int n){
ANS:       int i = 1;
ANS:       int col = 1;
ANS:       while ( i<=n*n){
ANS:           System.out.printf ( "%2d ", i);
ANS:           i++;
ANS:           if ( col == n){
ANS:               System.out.println( );
ANS:               col=1;
ANS:           }
ANS:           else
ANS:               col = col+1;
ANS:       }}

```

Question 4. Implementing a class

[15 marks]

The `DiaryPage` class is intended to represent pages from a diary or journal. A diary page should have a date and a title, and some text that is the entry for that day.

For example, a `DiaryPage` object with a date of the 3rd October 2006 and a title of “My Bicycle” and two lines of text might be printed out as follows:

```

date:  3/Oct/2006
title: My Bicycle
-----
Today, my bicycle broke.
When I stopped at the lights, the wheel fell off.
-----

```

Complete the `DiaryPage` class on the facing page.

The `DiaryPage` class should have fields to store a date, a title, and the text of the entry, a constructor, and four methods.

- The constructor should have three parameters: the day, month, and year, and should set the date of the page.
- `getDate()` should return the date in the form of a string. This method is already completed for you, and contains hints for representing the date.
- `setTitle(String t)` should set the title of the page. The title on a page can be set only once — `setTitle` should not change the title again once it has been set.
- `addToEntry(String txt)` should add a `String` to the current entry on the page. Each new string added to the entry should be start on a new line when it is printed out.
- `toString()` should construct a `String` representation of the page in the form shown in the example above, with the date, the title, and lines (made of `-`'s) before and after the entry.

The `DiaryPage` printed above could have been constructed with the calls

```

DiaryPage dp = new DiaryPage(3, "Oct", 2006);
dp.setTitle("My Bicycle");
dp.addToEntry("Today, my bicycle broke.");
dp.addToEntry("When I stopped at the lights, the wheel fell off.");
System.out.println(dp.toString ());

```

(Question 4 continued on next page)

(Question 4 continued)

```

/** Represents the information on a page of a diary */
public class DiaryPage{
    // fields
    ANS: private int day;
    ANS: private String month;
    ANS: private int year;
    ANS: private String title ;
    ANS: private String entry;
    ANS: OR private ArrayList<String> entry
    ANS: OR private String[] entry;   int entries;
    /* Constructor */
    public DiaryPage(   int d, String m, int y           ){
    ANS:     day = d;
    ANS:     month = m;
    ANS:     year =y;

    }
    public String getDate(){
        return (day+" / "+month+" / "+year);
    }

    public void setTitle (String t){
    ANS:     if ( title == null)
    ANS:         title = t;

    }
    public void addToEntry(String txt){
    ANS:     if (entry == null)
    ANS:         entry = e;
    ANS:     else
    ANS:         entry = entry + "\n"+e;

    ANS: OR   entry[entries++] = e;   //( if using array)
    ANS: OR   entry.add(e);           //( if using ArrayList)
    }
    public String toString(){
    ANS:     String ans = "date: " + getDate()+"\ntitle: "+ title;
    ANS:     return ans + "\n-----\n"+entry+"\n-----\n";

    ANS: OR for ( int j=0;j<entries;j++) ans=ans+"\n"+entry[j]; //( if using array)
    ANS: OR for (String e : entry) ans=ans+"\n"+e;           //( if using ArrayList)
    }
}

```

Question 5. Arrays

[36 marks]

Suppose you own several copies of a very popular CD. You agree to lend copies to some of your friends, and you want to implement a program to keep track of who has borrowed copies of the CD. You are prepared to let people borrow more than one copy of the CD (one for home, one for the car, etc.), but you want to place a limit on the number that any person can borrow.

The following is an outline of a class called `Loans`, to be used as part of this program. This class records who has borrowed copies of your CD using an array of strings called `names`. The `Loans` constructor takes two integer arguments, the first is the number of copies of the CD that you own, and the second is the limit on the number of copies that any person can borrow.

The outline contains incomplete declarations for the constructor and four methods, which you are required to implement.

```
public class Loans {
    private String[] names;
    private int numNames;
    private int limit ;
    public Loans(int numCDs, int lim) { }
    public void printLoans() { }
    public void borrowItem(String who) { }
    public void returnItem(String who) { }
    public void printMultiLoans() { }
}
```

(a) [4 marks] Complete the `Loans` constructor so that it performs any necessary initialisation.

```
public Loans(int numCDs, int lim) {
ANS:    names = new String[numCDs];
ANS:    numNames = 0;
ANS:    limit = lim;

}
```

(Question 5 continued on next page)

(Question 5 continued)

(b) [4 marks] Complete the `printLoans` method so that it prints a list of all the people who have borrowed your CD. If someone has borrowed more than one copy, their name should be printed once for each copy they have borrowed.

```
public void printLoans() {
```

```
ANS:     for ( int i = 0; i < numNames; i++) {
```

```
ANS:         System.out.println(names[i]);
```

```
ANS:     }
```

```
}
```

(c) [8 marks] Complete the `borrowItem` method so that it records that `who` has borrowed a copy of your CD, by adding his/her name to `names` — provided that he/she has not already borrowed the maximum number allowed. Print a suitable message if `who` has already reached the borrowing limit.

```
public void borrowItem(String who) {
```

```
ANS:     int c = 0;
```

```
ANS:     for ( int i = 0; i < numNames; i++)
```

```
ANS:         if ( name.equals(names[i]) )
```

```
ANS:             c++;
```

```
ANS:     if ( c >= limit ) {
```

```
ANS:         System.out.println("Too many loans");
```

```
ANS:         return;
```

```
ANS:     }
```

```
ANS:     names[numNames] = who;
```

```
ANS:     numNames++;
```

```
}
```

(Question 5 continued on next page)

(Question 5 continued)

(d) [8 marks] Complete the `returnItem` method so that it records that `who` has returned a copy of your CD, by removing one occurrence of his/her name from `names`. Print a suitable message if `who` does not currently have any copies of your CD.

```
public void returnItem(String who) {
```

```
ANS:   for (int i = 0; i < numNames; i++) {  
ANS:       if ( names[i].equals(who) ) {  
ANS:           numNames--;  
ANS:           names[i] = names[numNames];  
ANS:           return;  
ANS:       }  
ANS:   }  
ANS:   System.out.println("No loan found");
```

```
}
```

(Question 5 continued on next page)

(Question 5 continued)

(e) [12 marks] Complete the `printMultiLoans` method so that it prints a list of all the people who currently have two or more copies of your CD, along with the number of copies that each such person has.

To get full marks, you should ensure that no person is printed more than once in this list. You may define additional methods.

```
public void printMultiLoans() {
```

```
ANS:     String[] seen = new String[names.length];
ANS:     int n = 0;
ANS:     for ( int i = 0; i < numNames; i++) {
ANS:         String name = names[i];
ANS:         if ( count(name, names, numNames) > 1 && count(name, seen, n) == 0 ) {
ANS:             seen[n] = name;
ANS:             n++;
ANS:             System.out.println(name);
ANS:         }
ANS:     }
ANS: }
```

```
ANS: private int count(String name, String[] names, int n) {
ANS:     int c = 0;
ANS:     for ( int i = 0; i < n; i++)
ANS:         if ( name.equals(names[i]) )
ANS:             c++;
ANS:     return c;
ANS: }
```

Question 6. Recursion

[30 marks]

(a) [6 marks] Consider the following recursive method.

```
public int fn(int x, int y){
    if ( x == 1 )
        return y;
    else
        return fn(x/2, y+1);
}
```

For each of the following calls, show the sequence of calls performed and the final value that will be returned.

(i) [2 marks] `fn(1, 0);`

```
| fn(1, 0)
| => 0
| Output: 0
```

(ii) [2 marks] `fn(2, 0);`

```
| fn(2, 0)
|   fn(1, 1)
|   => 1
| => 1
| Output: 1
```

(Question 6 continued on next page)

(Question 6 continued)**(iii)** [2 marks] `fn(13, 0);`

```

fn(13, 0)
  fn(6, 1)
    fn(3, 2)
      fn(1, 3)
        => 3
      => 3
    => 3
  => 3
Output: 3

```

(b) [8 marks] Write an equivalent version of `fn` in part (a), using a loop instead of recursion.

```

public int fn(int x, int y) {
ANS:   while ( x > 1 ) {
ANS:     x = x/2;
ANS:     y = y+1;
ANS:   }
ANS:   return y;
}

```

(Question 6 continued on next page)

(Question 6 continued)

(c) [6 marks] Consider the following recursive method.

```
public void rec(String str) {
    int n = str.length();
    if ( n <= 2 ) {
        System.out.print(str);
    } else {
        rec(str.substring(n/2, n));
        rec(str.substring(0, n/2));
    }
}
```

For each of the following calls, show the sequence of calls performed and the output that is produced.

(i) [3 marks] `rec("abcd")`;

```
Rec (abcd)
  Rec (cd)
    Rec (ab)
Output:  cdab
```

(ii) [3 marks] `rec("recursion")`;

```
Rec (recursion)
  Rec (rsion)
    Rec (ion)
      Rec (on)
        Rec (i)
          Rec (rs)
            Rec (recu)
              Rec (cu)
                Rec (re)
Output:  onirscure
```

(Question 6 continued on next page)

(Question 6 continued)

(d) [10 marks] Consider the following method.

```

public boolean pal(int[] a) {
    int i = 0;
    int j = a.length-1;
    while ( i < j ) {
        if ( a[i] == a[j] ) {
            i++;
            j--;
        } else {
            return false;
        }
    }
    return true;
}

```

Write an equivalent pal method, using recursion instead of a loop.

Hint: You will need to define a recursive helper method.

```

ANS: public boolean pal(int[] a) {
ANS:     return pal(a, 0, a.length-1);
ANS: }
ANS:
ANS: public boolean pal(int[] a, int i, int j) {
ANS:     if ( i >= j ) return true;
ANS:     else if ( a[i] != a[j] ) return false;
ANS:     else return pal(a, i+1, j-1);
ANS: }

```

Question 7. Inheritance

[37 marks]

(a) [8 marks]

Consider the following code for two classes, `Adder` and `Badder`, and the tester method which uses these classes.

```
public class Adder {
    protected int x = 2;
    public Adder() {
        display ();
    }
    public void inc() {
        x++;
        display ();
    }
    public void dec() {
        x--;
        display ();
    }
    protected void display() {
        System.out.print(x + " ");
    }
}

public class Badder extends Adder {
    public void dec() {
        x=x-2;
        display ();
    }
}
```

```
public static void tester () {

    Adder a = new Adder();
    a.inc ();
    a.dec();
    a.inc ();
    a.dec();
    System.out.println ();

    Adder b = new Badder();
    b.inc ();
    b.dec();
    b.inc ();
    b.dec();
    System.out.println ();

}
```

What will `tester` print out when it is executed?

```
| 2 3 2 3 2
| 2 3 1 2 0
```

(Question 7 continued on next page)

(Question 7 continued)**(b)** [11 marks]

Consider the following class declarations:

```
class A {  
  
    public String eats() {  
        return "anything";  
    }  
  
    public String says() {  
        return "nothing";  
    }  
}  
  
class B extends A {  
  
    public String eats() {  
        return "grass";  
    }  
}  
  
class C extends A {  
  
    public String says() {  
        return "woof";  
    }  
}  
  
class D extends B {  
  
    public String says() {  
        return "baa";  
    }  
}
```

(Question 7 continued on next page)

(Question 7 continued)

Show the output that will be produced by each of the following code fragments.

(i) [2 marks]

```
A animal = new A();  
System.out.println("Eats " + animal.eats() + " and says " + animal.says());
```

```
| Eats anything and says nothing
```

(ii) [3 marks]

```
A animal = new B();  
System.out.println("Eats " + animal.eats() + " and says " + animal.says());
```

```
| Eats grass and says nothing
```

(iii) [3 marks]

```
A animal = new C();  
System.out.println("Eats " + animal.eats() + " and says " + animal.says());
```

```
| Eats anything and says woof
```

(iv) [3 marks]

```
A animal = new D();  
System.out.println("Eats " + animal.eats() + " and says " + animal.says());
```

```
| Eats grass and says baa
```

(Question 7 continued on next page)

(Question 7 continued)**(c)** [18 marks]

A computer system for a carrier company uses a class called `CarrierOrder` to store information about orders for goods to be transported and to compute the cost for such orders. As shown below, the class includes a constructor which takes as arguments the volume and weight of the goods to be transported and the distance they are to be taken. It also includes a method called `cost` which computes the cost for the order, and a method called `printOrder` which prints details of the order.

```

class CarrierOrder {

    public CarrierOrder(double vol, double wgt, double dist) {
    }

    public double cost() {
    }

    public void printOrder() {
    }

    ...
}

```

The carrier company decides to offer an express service, for which they guarantee same day delivery. This service is only available for orders upto a specified maximum volume, weight and distance, and the price is 50% greater than the normal price for the same order. Express orders that do exceed the maximum volume, weight or distance are treated like ordinary orders and charged at the normal rate.

Express orders will be handled by a new `ExpressOrder` class which should be a subclass of `CarrierOrder`. An outline of the `ExpressOrder` class is shown opposite. The maximum volume, weight and distance for an express order are specified in `static` fields.

You are required to implement the `ExpressOrder` constructor, and make whatever other additions are required so that the `cost` and `printOrder` methods work correctly for `ExpressOrder` objects.

The details printed by `printOrder` should be exactly the same as is printed for a standard order.

You must not modify `CarrierOrder` in any way. You do not need to know what fields `CarrierOrder` has, how these methods are defined, or what other methods `CarrierOrder` has, except that they cannot change the volume, weight or distance of the order.

(Question 7 continued on next page)

(Question 7 continued)

```

class ExpressOrder    ANS: extends CarrierOrder    {

    private static double maxVol = ... ;
    private static double maxWgt = ... ;
    private static double maxDist = ... ;
ANS: private Boolean express;

    public ExpressOrder(double vol, double wgt, double dist){
ANS:         super(vol, wgt, dist );
ANS:         express = ( vol<=maxVol && wgt <= maxWgt && dist <= maxDist);

    }

ANS: public double cost() {
ANS:     double ans = super.cost();
ANS:     if ( express )
ANS:         ans = ans * 1.5;
ANS:     return ans;
ANS: }

ANS: // Note, this is not the only way of solving this problem.
ANS: // You could also store the volume, weight, and distance in the ExpressOrder object ,
ANS: // and determine whether the 1.5 multiplier should be applied inside the cost method.

}

```
