

**UNIVERSITY COLLEGE LONDON**

*University of London*

**EXAMINATION FOR INTERNAL STUDENTS**

*For The Following Qualification:-*

*B.Sc.*

**Information Studs. G4: Programming 1**

**COURSE CODE : INSTG004**

**UNIT VALUE : 0.50**

**DATE : 22-MAY-03**

**TIME : 14.30**

**TIME ALLOWED : 2 Hours**

# INSTG004 (“Programming 1”), 2003

Answer **ANY THREE** questions.

## Note to candidates:

Throughout this exam paper you may assume the existence of, and use, all of the methods in the *Keyboard* class, as described in the course textbook, for user input in programs. In particular you may use the following methods, all of which return a value of the indicated type obtained from user keyboard input:

```
static boolean readBoolean ()
static byte readByte ()
static char readChar ()
static double readDouble ()
static float readFloat ()
static int readInt ()
static long readLong ()
static short readShort ()
static String readString ()
```

**PAPER CONTINUED ON NEXT PAGE**

1. a) Briefly explain what is meant by *Java bytecode*, and why Java is described as an *architecture neutral* (or *cross-platform*) programming language. [4 marks]
- b) Briefly explain what are meant by *compilation errors*, *runtime errors* and *logical errors* in computer programming. [3 marks]
- c) Briefly explain the terms *variable*, *constant*, and *data type* in the context of the Java programming language. [3 marks]
- d) A certain museum currently charges a full price entrance fee of £12. However, it lets students and pensioners in for half price, and lets people who are both students and pensioners in for a quarter of the full price. Use an appropriate combination of “if ... else ...” statements to complete the following program, which outputs the entrance fee for a single visitor. The program should output “Incorrect input”, “Your ticket costs 12 pounds”, “Your ticket costs 3 pounds” or “Your ticket costs 6 pounds”. You may assume that the user enters a single lower case character after each prompt. Your solution should make it as easy as possible to alter the full price entrance fee at a later date, but you may assume that the full price will always be a multiple of 4 (so that a quarter of the price will always be a whole number).

```
import cs1.Keyboard;
public class EntranceFeeCalculator
{
    public static void main(String[] args)
    {
        final int FULL_PRICE = 12;
        System.out.print("Are you a student? (y/n): ");
        char isStudent = Keyboard.readChar();
        System.out.print("Are you a pensioner? (y/n): ");
        char isPensioner = Keyboard.readChar();

        add an appropriate combination of "if ... else ..." statements here
    }
}
```

[10 marks]

[Total 20 marks]

**PAPER CONTINUED ON NEXT PAGE**

2. a) Briefly explain what is meant by the *flow of control* in a Java program. [2 marks]
- b) Re-write the following program fragment using a while loop instead of a for loop.

```
for (int count = 1; count <= MAX_NO_OF_UNITS; count++)
{
    System.out.print("\t\t" + count + "\t\t");
    System.out.println((count * UNIT_PRICE) - DISCOUNT);
}
```

[3 marks]

- c) A certain music shop is selling CDs for £11 each if the customer buys 4 or less CDs, or £9 each if the customer buys 5 or more CDs. A program is being written that will allow the user to enter a minimum and maximum number of CDs and then produce a table of prices, as in the following example input/output:

```
Enter minimum number of CDs: 4
Enter maximum number of CDs: 6
```

CDs	PRICE
4	44
5	45
6	54

The main method of the program begins as follows:

```
public static void main(String[] args)
{
    final int FULL_PRICE = 11, REDUCED_PRICE = 9;
    final int NO_FOR_DISCOUNT = 5;
    System.out.print("\n\tEnter minimum number of CDs: ");
    int minimum = Keyboard.readInt();
    System.out.print("\tEnter maximum number of CDs: ");
    int maximum = Keyboard.readInt();
    .....
```

Use a “for” or “while” loop to complete the main method of the program so that it outputs the table correctly. For this part of the question you may assume that the user enters a whole number between 1 and 5 after the first prompt and a whole number between 6 and 10 for the second prompt, so that it is not necessary to include any input error checking in the program.

[6 marks]

- d) Using “while” loops, modify the given program fragment in part (c) so that it performs input error checking. The error checking should ensure that the table of prices has a last entry of 50 or less CDs, and that the table is no more than 10 lines long (excluding the table headings). For this part of the question you may assume only that the user always enters a (positive or negative) whole number at every prompt.

[9 marks]

[Total 20 marks]

**PAPER CONTINUED ON NEXT PAGE**

3. a) Briefly explain what a *constructor method* is in the context of a Java class definition. [4 marks]
- b) Briefly explain the use of the *private* and *public* modifiers in the context of the Java programming language. [4 marks]
- c) A machine part called a “flow regulator tube” is to be represented as a Java class. Flow regulator tubes have an “aperture” that can be set to any whole number between 1 and 5 inclusive. They also have a “safety valve” that should automatically be set to “open” if the aperture setting is more than 3, and “closed” if the aperture setting is less than 2. (If the aperture is set to 2 or 3 the safety valve can be left either “open” or “closed”.) Write a class definition for flow regulator tubes. Your definition should be such that, when tested with the following program,

```
public class FlowRegulatorTubeTestProg
{
    public static void main(String[] args)
    {
        FlowRegulatorTube aTube = new FlowRegulatorTube();
        char selection = 'a';
        while (selection != 'q')
        {
            System.out.println("Type: s - to set the aperture");
            System.out.println("      v - to view the tube status");
            System.out.println("      q - to quit");
            System.out.print("Enter selection: ");
            selection = Keyboard.readChar();
            if (selection == 's')
            {
                System.out.print("Enter new aperture setting: ");
                int setting = Keyboard.readInt();
                aTube.setAperture(setting);
            }
            if (selection == 'v')
                System.out.println(aTube.toString());
        }
        System.out.println("PROGRAM ENDED.");
    }
}
```

the following example input/output can be reproduced:

```
Type: s - to set the aperture
      v - to view the current tube status
      q - to quit
Enter selection: s
Enter new aperture setting: 5
Type: s - to set the aperture
      v - to view the current tube status
      q - to quit
Enter selection: v
      Aperture: 5, safety valve: open.
Type: s - to set the aperture
      v - to view the current tube status
      q - to quit
Enter selection: q
PROGRAM ENDED.
```

[12 marks]

[Total 20 marks]

**PAPER CONTINUED ON NEXT PAGE**

4. a) Briefly explain what is meant by *method decomposition* in the context of Java programming. [3 marks]
- b) Briefly explain the difference between *primitive data variables* and *object reference variables* in the context of Java programming. [4 marks]
- c) A certain D.I.Y. shop sells small nails only in packets of 50, and large nails only in packets of 40. A packet of small nails costs £2, and a packet of large nails costs £3. Customers who buy 1000 or more nails are offered a discount of £5. So as not to run out of nails, the shop refuses to sell more than 5000 nails to any one customer. The following Java program allows customers to enter the amount of small and large nails they want to buy and find out what the total cost will be.

```
import cs1.Keyboard;
public class NailCostCalculator
{
    public static final int LARGE_PER_PACK = 40, SMALL_PER_PACK = 50;
    public static final int LARGE_PACK_COST = 3, SMALL_PACK_COST = 2;
    public static final int MAX_NAILS = 5000, NO_FOR_DISCOUNT = 1000;
    public static final int DISCOUNT = 5;

    public static void main(String[] args)
    {
        int smallNails, largeNails, smallCost, largeCost, totalCost;
        System.out.print("Enter number of small nails: ");
        smallNails = Keyboard.readInt();
        System.out.print("Enter number of large nails: ");
        largeNails = Keyboard.readInt();
        if (smallNails >= 0 && (smallNails % SMALL_PER_PACK == 0)
            && largeNails >= 0 && (largeNails % LARGE_PER_PACK == 0)
            && (smallNails + largeNails <= MAX_NAILS))
        {
            smallCost = (smallNails/SMALL_PER_PACK) * SMALL_PACK_COST;
            largeCost = (largeNails/LARGE_PER_PACK) * LARGE_PACK_COST;
            totalCost = smallCost + largeCost;
            if (smallNails + largeNails >= NO_FOR_DISCOUNT)
                totalCost -= 5;
            System.out.print("The total cost is ");
            System.out.println(totalCost + " pounds.");
        }
        else
            System.out.println("This combination is not possible.");
    }
}
```

This program is to be re-written so that the main program in the new version is as follows:

```
public static void main(String[] args)
{
    int smallNails, largeNails;
    System.out.print("Enter number of small nails: ");
    smallNails = Keyboard.readInt();
    System.out.print("Enter number of large nails: ");
    largeNails = Keyboard.readInt();
    if (amountsOK(smallNails, largeNails))
        outputPrice(smallNails, largeNails);
    else
        System.out.println("This combination is not possible.");
}
```

P.T.O.

QUESTION 4 CONTINUED ON NEXT PAGE

Write definitions for the methods “amountsOK(…)” and “outputPrice(…)” (both of which are invoked in the “if” part of the “if...else...” statement in this new version) so that the new program has identical output (for any given set of inputs) to the original.

[7 marks]

- d) Briefly explain why it is not possible to write a method “inputAmounts(…)” such that the main program in part (c) can be re-written as follows:

```
public static void main(String[] args)
{
    int smallNails, largeNails;
    inputAmounts(smallNails, largeNails);
    if (amountsOK(smallNails, largeNails))
        outputPrice(smallNails, largeNails);
    else
        System.out.println("This combination is not possible.");
}
```

and explain what other change or changes would need to be made to the program to make any definition of “inputAmounts(…)” work properly.

[6 marks]

[Total 20 marks]

**PAPER CONTINUED ON NEXT PAGE**

5. This question concerns a particular type of ticket machine used in theatres. Such ticket machines can issue a maximum of 500 tickets per day, and can either issue a ticket to a child for £2, or to an adult for £5. This type of ticket machine is therefore represented as a Java class "TicketMachine" as follows:

```
public class TicketMachine
{
    public static final int CHILD_PRICE = 2, ADULT_PRICE = 5;
    public static final int MAX_TICKETS_PER_MACHINE = 500;

    private int childTicketsSold;
    private int adultTicketsSold;

    //-----
    //  Constructor - creates a new ticket machine.
    //-----
    public TicketMachine ()
    {
        childTicketsSold = 0;
        adultTicketsSold = 0;
    }

    //-----
    //  Sets the number of child tickets sold.
    //-----
    public void setChildTicketsSold (int number)
    {
        if (number < 0)
            childTicketsSold = 0;
        else if (number + adultTicketsSold <= MAX_TICKETS_PER_MACHINE)
            childTicketsSold = number;
        else
            childTicketsSold = MAX_TICKETS_PER_MACHINE - adultTicketsSold;
    }

    //-----
    //  Sets the number of adult tickets sold.
    //-----
    public void setAdultTicketsSold (int number)
    {
        if (number < 0)
            adultTicketsSold = 0;
        else if (number + childTicketsSold <= MAX_TICKETS_PER_MACHINE)
            adultTicketsSold = number;
        else
            adultTicketsSold = MAX_TICKETS_PER_MACHINE - childTicketsSold;
    }

    //-----
    //  Returns the number of child tickets sold.
    //-----
    public int getChildTicketsSold ()
    {
        return childTicketsSold;
    }

    //-----
    //  Returns the number of adult tickets sold.
    //-----
    public int getAdultTicketsSold ()
    {
        return adultTicketsSold;
    }
}
```

P.T.O.

QUESTION 5 CONTINUED ON NEXT PAGE



- a) Briefly explain what an *array*, an *array element* and an *array index* are in the context of Java programming. [4 marks]
- b) A particular theatre has 6 ticket machines of the type described above. A Java program is to be written to keep track of the number of tickets the theatre sells using these machines each day. The program utilises the “TicketMachine” class described above and begins as follows:

```
public class TicketsSoldRecorder
{
    public static void main(String[] args)
    {
        final int NO_OF_MACHINES = 6;
        TicketMachine[] ticketMachineList = new TicketMachine[NO_OF_MACHINES];
        for (int count = 0; count < NO_OF_MACHINES; count++)
            ticketMachineList[count] = new TicketMachine();
        .....
```

Without adding any input error checking in addition to that already incorporated in the “TicketMachine” class, extend the main method of this program so that the user is able to enter the number of child and adult tickets each machine has issued, as in the following example input/output:

```
Enter child tickets sold for machine 1: 134
Enter adult tickets sold for machine 1: 65
Enter child tickets sold for machine 2: 78
Enter adult tickets sold for machine 2: 95
Enter child tickets sold for machine 3: 23
Enter adult tickets sold for machine 3: 90
Enter child tickets sold for machine 4: 290
Enter adult tickets sold for machine 4: 108
Enter child tickets sold for machine 5: 45
Enter adult tickets sold for machine 5: 77
Enter child tickets sold for machine 6: 31
Enter adult tickets sold for machine 6: 28
```

[6 marks]

- c) Further extend the main method of the program in part (b) so that after the user has entered the numbers of tickets issued from each machine, the program is able to output the total cost of all the tickets issued, and report which machine has issued the most number of tickets, as in the example input/output below. For this part of the question you may assume that no two machines have issued the same number of tickets. It is not necessary to write out your code for part (b) again when answering this part of the question.

```
Enter child tickets sold for machine 1: 134
.....
.....
Enter adult tickets sold for machine 6: 28
The total cost of all the tickets sold is 926 pounds.
Ticket machine number 4 sold the most tickets.
```

[10 marks]

[Total 20 marks]

**END OF PAPER**