UNIVERSITY COLLEGE LONDON

University of London

EXAMINATION FOR INTERNAL STUDENTS

For the following qualifications:-

B.Sc.

Information Studs. G4: Programming 1

COURSE CODE	:	**INSTG004**
DATE	:	**11-May-05**
TIME	:	**14:30**
TIME ALLOWED	:	**2 hours 0 minutes**

INSTG004 (Programming 1) 2005

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 and used in the course, 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 ()
```

1. a) Give two examples of *data types* in Java. Give an example of how to convert a piece of data from one type to another.

[3 marks]

b) Briefly explain the difference between the use of a single "=" sign in Java and the use of a double "==" sign.

[2 marks]

c) What is meant by a *Boolean expression* in Java? What truth values (`true` or `false`) do the following expressions evaluate to?

```
(2 <= 7 || 4 < 9) && 5 > 6

2 <= 7 || (4 < 9 && 5 > 6)
```

[5 marks]

d) A certain theme park has the following entrance price policy. In the summer season people age 16 or over pay the full price, and people over 3 years old but under 16 years old pay half price. In the winter season everybody over 3 years old pays half price. People 3 years old or under get in free all year round. The current full price is £12. Use an appropriate combination of "if ... else ... " statements to complete the following program, which outputs the entrance price for a given user. The program should output "Incorrect input", "You may enter free of charge", "Please pay 6 pounds" or "Please pay 12 pounds". You may assume that the user enters a single lower case character after the first prompt, and a positive whole number after the second prompt. You may also assume that the full price in pounds will always be an even whole number.

```
import cs1.Keyboard;
public class ParkEntrance
{
  public static void main(String[] args)
  {
    final int FULL_PRICE = 12, MAX_POSS_AGE = 130;
    System.out.print("Enter 'w' for winter, 's' for summer: ");
    char season = Keyboard.readChar();
    System.out.print("Enter your age in years: ");
    int age = Keyboard.readInt();

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

  }
}
```

[10 marks]

[Total 20 marks]

2. a) Briefly explain what is meant by the *scope* of a variable in a Java program.

[3 marks]

b) Re-write the following program fragment using a for loop instead of a while loop.

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

[2 marks]

c) A certain video shop is selling DVDs for £13 each if the customer buys 3 or less DVDs, or £11 each if the customer buys 4 or more DVDs. A program is being written that will allow the user to enter a minimum and maximum number of DVDs and then produce a table of prices, as in the following example input/output:

```
Enter minimum number of DVDs: 3
Enter maximum number of DVDs: 5

DVDs            PRICE
 3               39
 4               44
 5               55
```

The main method of the program begins a follows:

```
public static void main(String[] args)
{
  final int FULL_PRICE = 13, REDUCED_PRICE = 11;
  final int NO_FOR_DISCOUNT = 4;
  System.out.print("\n\tEnter minimum number of DVDs: ");
  int minimum = Keyboard.readInt();
  System.out.print("\tEnter maximum number of DVDs: ");
  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 4 after the first prompt and a whole number between 5 and 8 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 DVDs, 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]

3. a) Should a *constructor method* in a Java class definition be *private* or *public*? Briefly explain your answer.

[6 marks]

b) A type of boiler that can have its temperature set and its pump turned off and on is to be represented as a Java class. The boiler must always operate within the following safety constraints: (i) the temperature must never be less than 10 or more than 80 (degrees Celsius), and (ii) if the pump is turned off the temperature must not be set to more than 50. Write a class definition for the boiler. Your definition should be such that, when tested with the following program,

```
import cs1.Keyboard;
public class BoilerTestProg
{
  public static void main(String[] args)
  {
    Boiler aBoiler = new Boiler();
    System.out.println(aBoiler.toString());
    char selection = 'a';
    while (selection != 'q')
    {
      System.out.print("\nType:  ");
      System.out.println("\tp - to turn on pump");
      System.out.println("\to - to turn off pump");
      System.out.println("\tt - to set temperature");
      System.out.println("\tq - to quit\n");
      System.out.print("\tEnter selection: ");
      selection = Keyboard.readChar();
      switch (selection)
      {
        case 'p':
          aBoiler.turnOnPump();
          System.out.println(aBoiler.toString());
          break;
        case 'o':
          aBoiler.turnOffPump();
          System.out.println(aBoiler.toString());
          break;
        case 't':
          System.out.print("\tEnter new temperature: ");
          int temperature = Keyboard.readInt();
          aBoiler.setTemperature(temperature);
          System.out.println(aBoiler.toString());
          break;
      }
    }
    System.out.println("\tPROGRAM ENDED\n");
  }
}
```

the following example input/output can be reproduced:

```
         Status: Temperature 10,  Pump off

Type:   p - to turn on pump
        o - to turn off pump
        t - to set temperature
        q - to quit

        Enter selection: p
         Status: Temperature 10,  Pump on
```

P.T.O.

```
Type:   p - to turn on pump
        o - to turn off pump
        t - to set temperature
        q - to quit

        Enter selection: t
        Enter new temperature: 60
        Status: Temperature 60,  Pump on

Type:   p - to turn on pump
        o - to turn off pump
        t - to set temperature
        q - to quit

        Enter selection: o
        Status: Temperature 50,  Pump off

Type:   p - to turn on pump
        o - to turn off pump
        t - to set temperature
        q - to quit

        Enter selection: t
        Enter new temperature: 60
        Status: Temperature 50,  Pump off

Type:   p - to turn on pump
        o - to turn off pump
        t - to set temperature
        q - to quit

        Enter selection: q
        PROGRAM ENDED
```

(The numbers and letters in bold are user inputs.)

[14 marks]

[Total 20 marks]

4. This question uses the following Java class “WholeNum”:

```
public class WholeNum
{
   private int value;

   //-----------------------------------------------------------------
   //  Constructor - sets value to value of the parameter.
   //-----------------------------------------------------------------
   public WholeNum (int initialValue)
   {
      value = initialValue;
   }

   //-----------------------------------------------------------------
   //  Returns the current value.
   //-----------------------------------------------------------------
   public int getValue ()
   {
      return value;
   }

   //-----------------------------------------------------------------
   //  Sets the value to the value of the parameter.
   //-----------------------------------------------------------------
   public void setValue (int newValue)
   {
      value = newValue;
   }
}
```

a) A 1 litre can of a certain type of paint can cover 3 square meters of wall. The following program calculates how many cans of paint will be needed to cover a wall of a height and width given by the user:

```
import cs1.Keyboard;
public class PaintCalc
{
   public static final int COVERAGE = 3, MAX_METERS = 20;

   public static void main(String[] args)
   {
      WholeNum height = new WholeNum(0), width = new WholeNum(0);

      while (height.getValue() < 1 || height.getValue() > MAX_METERS)
      {
         System.out.print("How high is the wall in meters? ");
         height.setValue(Keyboard.readInt());
         if (height.getValue() < 1 || height.getValue() > MAX_METERS)
            System.out.print("ERROR! - ");
      }
      while (width.getValue() < 1 || width.getValue() > MAX_METERS)
      {
         System.out.print("How wide is the wall in meters? ");
         width.setValue(Keyboard.readInt());
         if (width.getValue() < 1 || width.getValue() > MAX_METERS)
            System.out.print("ERROR! - ");
      }
      int area = height.getValue() * width.getValue();
      System.out.print("Number of cans needed: ");
      if (area % COVERAGE == 0)
         System.out.println(area / COVERAGE);
      else
         System.out.println((area / COVERAGE) + 1);
   }
}
```

P.T.O.

Rewrite this program using method decomposition, ensuring that the output remains exactly the same. The first two methods of the program should be exactly as follows:

```
public static void main(String[] args)
{
  WholeNum height = new WholeNum(0), width = new WholeNum(0);
  inputDimensions(height, width);
  outputCansNeeded(height, width);
}

public static boolean metersOK(int aMeasurement)
{
  return (aMeasurement >= 1 && aMeasurement <= MAX_METERS);
}
```

Your answer should therefore include appropriate definitions for "`inputDimensions(…)`" and "`outputCansNeeded(…)`", and should make appropriate use of the method "`metersOK(…)`" above.

[8 marks]

b) Briefly explain why it is necessary to use instances of the "WholeNum" class (rather than primitive data variables of type "int") in order to successfully implement the method decomposition in part (a).

[4 marks]

c) A certain bureau de change is selling euros at 1.5 euros to the pound, but for amounts up to and including £40 first takes off £4 commission. Therefore (for example) £24 buys 30 euros, and £45 buys 67.50 euros. Write a public static method "`inEuros(...)`" which takes an amount in pounds as an argument and returns the amount that the bureau de change will give in euros. For example, the call `inEuros(24)` should return `30`, and the call `inEuros(45)` should return `67.5`.

[6 marks]

d) Briefly comment on how the method you have defined in part (c) could be modified so that it could be used for various different commission charges and exchange rates.

[2 marks]

[Total 20 marks]

5. This question is about a program for a car retailer that stores the last names of its sales representatives and the number of cars they have each sold during the current week. The question and program assume the existence of a class "SalesRep" with the following public methods:

```
public SalesRep ()
public void setRepName (String aName)
public void setRepsCarsSold (int aNumber)
public String getRepName ()
public int getRepsCarsSold ()
```

The first of these methods is a constructor which sets an initial value of "none" for the sales representative name and –1 for the number of cars sold (to indicate an "unused" SalesRep instance). The other methods are standard "set" and "get" methods for the two items of data associated with each sales representative.

In the program, the sales representative information is stored in an array of SalesRep object instances called "`salesRepList`" which is declared in the "main(…)" method. The main menu of the program gives the user two options (apart from quitting the program). The first option is to enter all the sales representatives' details, and the second option is to display details of what bonuses will be awarded that week. The first part of the program is as follows:

```
import cs1.Keyboard;
public class SalesRepProg
{
  public static final int MAX_NO_OF_REPS = 50;
  public static final int MAX_POSS_NO_OF_CARS_SOLD = 100;
  public static final int MIN_NO_FOR_BONUS = 15;
  public static final int BONUS = 300;

  public static void main(String[] args)
  {
    SalesRep[] salesRepList = new SalesRep[MAX_NO_OF_REPS];
    for (int index = 0; index < salesRepList.length; index++)
      salesRepList[index] = new SalesRep();

    char option = 'z';
    while (option != 'q')
    {
      System.out.println("\nEnter:");
      System.out.println("   i - to input sales rep details");
      System.out.println("   b - to display sales rep bonuses");
      System.out.println("   q - to quit the program");
      System.out.print("Type your option and press RETURN: ");
      option = Keyboard.readChar();

      switch (option)
      {
        case 'i': inputDetails(salesRepList);
                  break;

        case 'b': displayBonuses(salesRepList);
                  break;
      }
    }
  }
```

P.T.O.

QUESTION 5 CONTINUED ON NEXT PAGE

a) After carefully examining the above program fragment, write a suitable definition for the method "`inputDetails(...)`" (which is called from inside the switch statement in the "`main(...)`" method). Invocation of this method via an input of "i" from the main user menu should cause the program to have input/output such as:

```
Please enter the total number of sales reps: 10

SALES REP NUMBER 1
Enter rep's last name: Jones
Enter number of cars sold by Jones: 27

SALES REP NUMBER 2
Enter rep's last name: Miller
Enter number of cars sold by Miller: 9

.............
.............
.............

SALES REP NUMBER 9
Enter rep's last name: Patel
Enter number of cars sold by Patel: 23

SALES REP NUMBER 10
Enter rep's last name: Smith
Enter number of cars sold by Smith: 13
```

(The names and numbers in bold are user inputs.)

[12 marks]

b) The car retailer rewards its sales personnel by giving a bonus of £300 to any representative who sells 15 or more cars. Write a definition for the method "`displayBonuses(...)`" (which is also called from inside the switch statement in the "`main(...)`" method). The method should cause the program to have input/output such as:

```
The following people will receive a 300 pound bonus:

      Jones
      Patel
```

[8 marks]

[Total 20 marks]