THIS PAPER IS NOT TO BE REMOVED FROM THE EXAMINATION HALLS

UNIVERSITY OF LONDON **CO1109 ZA**

BSc, CertHE and Diploma Examination

COMPUTING AND INFORMATION SYSTEMS and CREATIVE COMPUTING

Introduction to Java and Object-Oriented Programming

Date and Time: Wednesday 3 May 2017: 10:00 - 13:00

Duration: 3 hours

There are SIX questions on this paper. Candidates should answer **FOUR** questions. Full marks will be awarded for complete answers to **FOUR** questions. Each question is worth 25 marks. The mark for each part of a question is indicated in [ ] brackets.

Only your first **FOUR** answers, in the order that they appear in your answer book, will be marked.

There are 100 marks available on this paper.

No calculators should be used.

Question 1

(a)

(i) Consider class *Bool2:*

```
public class Bool2{

     public static void main(String[] args){
          boolean b = (3==3);
          System.out.println(b);
     }
}
```

Which one of the following is true about class *Bool2*? [2 marks]

(A) The class will not compile.

(B) The class will compile and output `true`

(C) The class will compile and output `false`

(D) The class will compile and output `3`

(E) The class will compile and output `4`

(ii) Consider class *Bool4:*

```
import java.util.Scanner;

public class Bool4{

     public static void main(String[] args){
          Scanner in =new Scanner(System.in);
          System.out.print("Enter Number>");
          int t=in.nextInt();
          if (t==0 && t==3) System.out.println("hello");
     }
}
```

Which of the following describes the behaviour of *Bool4* when the user enters 3? [3 marks]

(A) The program will output `hello`

(B) The program will output nothing at all.

(C) The program will output `0`

(D) The program will not compile.

(E) None of the above.

(iii) Write a Java class called *Silly* that will compile successfully and will run, but will **not** output anything. [3 marks]

(b) For each of the following loops, say how many asterisks (*) will be printed. If you think that the loop will continue indefinitely without stopping, write 'infinite loop' in your answer book for that part of the question. [9 marks]

(i)
```
for(int i=0;i<9;i++) System.out.print("*");
```

(ii)
```
int i=10; while (i>7){System.out.print("*");i--;}
```

(iii)
```
for(int i=-2;i<3;i++) System.out.print("*");
```

(iv)
```
int i=1; while (i>19){System.out.print("*");i=i+2;}
```

(v)
```
for(int i=1;i>=1;i++) System.out.print("*");
```

(vi)
```
int i=0; while (i<200){System.out.print("*");i=i+10;}
```

(vii)
```
for(int i=0;i<7;i=i+2) System.out.print("*");
```

(viii)
```
int i=0; while(i<5){System.out.print("*");i--;}
```

(ix)
```
for(int i=5; i>6; i++) System.out.print("**");
```

(c) Give the output of the following program. [8 marks]

```
public class ShapeA{
     public static void main(String[] args){
          for (int i = 0; i < 4; i++){
               for (int j =i; j<4; j++)System.out.print("*");
               System.out.println();
          }
          for (int i = 0; i < 3; i++) System.out.println("#");
          for (int i = 3; i >= 0; i--){
               for (int j = i; j<4; j++)System.out.print("*");
               System.out.println();
          }
     }
}//end of class
```

Question 2

(a) What are the syntax errors in the following Java classes?

(i) [2 marks]

```
public class W{

     public static void main(String[] args){
          int x;
          x = 3.142;
     }
}
```

(ii) [2 marks]

```
public class X{

     int k;

     public static void main(String[] args){
          k = 1;
     }
}
```

(iii) [2 marks]

```
public class Y{

     public static void main(String[] args){
               String s = hello;
     }
}
```

(iv) [2 marks]

```
public class Z{

     static add(int x, int y){
          return x + y;
     }

     public static void main(String[] args){
          int k = add(5, 19);
     }
}
```

(b) For each of the following expressions state whether it type checks correctly or not. Write 'yes' in your answer book if it type checks correctly, and write 'no' if it does not. [9 marks]

(i) `Math.abs("Excalibur");`

(ii) `Math.abs("Excalibur".length());`

(iii) `int x = Math.abs("Excalibur".length())+2016;`

(iv) `Integer.parseInt("401");`

(v) `"strange".compareTo("401");`

(vi) `"strange".charAt(401);`

(vii) `int z = "strange".compareTo("401")-51;`

(viii) `"break".replace('k',"doorstop".charAt(0)).length();`

(ix) `"break".replace('k',"doorstop".charAt(0).length());`

(c) Write the following methods:

(i) Write a static method *min(int, int)*, which takes two `int` parameters and returns the smallest of the two. [4 marks]

(ii) Write a static method *minOf3(int, int, int)* which calls *min(int, int)* above. [4 marks]
Your *minOf3(int, int, int)* method should take three `int` parameters and return the smallest of the three.

Question 3

(a) Consider the *HB* class, below:

```
import java.util.Random;

public class HB {

      static String[] answers = new String[]
      {"Without a doubt","Yes", "Concentrate and
      ask again", "Don't count on it",  "Very
      doubtful", "Outlook unclear"};

      public static void main(String[] args) {
            String m8ball1 = magic8Ball();
            System.out.println(m8ball1);
      }
}
```

You will note that the *HB* class will not compile as it is, since there is no method called *magic8ball()*. [8 marks]
Write the *magic8Ball()* method. The method returns a random element from the *answers* `Array`.

You may find the following extract from the Java API helpful:

Class Random

Constructor Detail

`public Random()`
Creates a new random number generator.

Method Detail

`public int nextInt(int n)`
The general contract of `nextInt` is that one `int` value in the specified range is pseudorandomly generated and returned. All `n` possible `int` values are produced with (approximately) equal probability.

Parameters:
`n` - the bound on the random number to be returned. Must be positive.

Returns:
the next pseudorandom, uniformly distributed `int` value between `0` (inclusive) and `n` (exclusive) from this random number generator's sequence

Throws:
`IllegalArgumentException` - if `n` is not positive

(b) Consider the *HelloX* class, below:

```
import java.util.Scanner;

public class HelloX {
     public static void main(String[] args) {
          Scanner in = new Scanner(System.in);
          int num = 0;
          String name = "";

          System.out.print("Enter a number> ");
          num = in.nextInt();

          System.out.print("What is your name? ");
          name = in.nextLine();

          System.out.println("Hello " + name + "! You
              entered the number " + num);

          in.close();
     }
}
```

(i) In the statement:

```
Scanner in = new Scanner(System.in);
```

which one of the following best describes the purpose of `System.in`? [3 marks]

(A) It tells the *Scanner* class to accept input from an object called 'in'.

(B) It associates the *Scanner* object with the keyboard.

(C) It links the *Scanner* object with the output to the screen.

(D) None of the above.

(ii) A user runs the above program, and when prompted for input, with `"Enter a number> "`, enters the number 5.9, then presses the return key. [3 marks]
Which one of the following best describes what is most likely to happen next?

(A) The program outputs `"Enter a number> "` again.

(B) The program ends with an exception.

(C) The program outputs `"What is your name?"`.

(D) None of the above.

(iii) A user called Fred compiles and runs the program, but once he has entered the number '5' and pressed the return key the program ends.
The program run is as follows:

```
Enter a number> 5
What is your name? Hello ! You entered the number 5
Press any key to continue . . .
```

Why did the program end before Fred could enter his name? [3 marks]

(A) The *nextInt()* method did not read the new line after the number 5.

(B) The *nextInt()* method closed the *Scanner* object.

(C) The *nextLine()* method does not belong to the *Scanner* class.

(D) None of the above.

(c)

```
public class Much {
   public static String what(String word) {
        int len = word.length();
        String hash = "";
        for (int i = 0; i < len; i++) {
             hash += "#";
        }
        return hash;
   }
   public static void print(String line) {
        System.out.println(line);
   }
   public static String[] dis(int x, int y) {
        String[] s = new String[y];
        String t = "";
        for (int i = 0; i < y; i++) {
             for (int j = 0; j < x; j++) {
                  t += "*";
             }
             s[i] = t;
             t = "";
        }
        return s;
   }
   public static void print(String[] lines) {
        for (int i = 0; i < lines.length; i++) {
             System.out.println(lines[i]);
        }
   }
   public static void main(String[] args) {
        String w1 = what("this");
        String w2 = what("THISTHAT");
        print("");
        print(w1);
        print(w2);
        print("");
        String[] c1 = dis(1, 1);
        String[] c2 = dis(4, 2);
        String[] c3 = dis(2, 4);
        print(c1);
        print("");
        print(c2);
        print("");
        print(c3);
        print("");
   }
}
```

Consider the *Much* class, above.

(i) Give the output of the class (assume that it compiles and runs successfully). [6 marks]

(ii) The *Much* class has two methods called *print*. Does this mean: [2 marks]

(A) The program will not compile because of a name clash with the two print methods.

(B) This is an example of method overriding and will not prevent the program from compiling and running.

(C) This is an example of method overloading and will not prevent the program from compiling and running.

(D) None of the above.

Question 4

(a) Consider the following class:

```
import java.io.*;
public class filey{

  public static void bling(String s) throws Exception{
     BufferedReader inone =new BufferedReader(new
     FileReader(s));
     int t=inone.read();
     while (t!=-1){
          System.out.print((char)t);
          t=inone.read();
     }
  }

  public static void main(String[] args) throws Exception{
     bling(args[0]);
  }
}
```

(i) What would be the output if we run the *filey* class with `filey.java` given as the input parameter? [2 marks]

(ii) What does it mean when the variable *t* gets the value -1? [2 marks]

(iii) What would happen if we ran *filey.java* with an empty input file? Why? [2 marks]

(iv) What would happen if we ran *filey.java* with an input file that did not exist? [2 marks]

(b) Consider the *SwapInts* class below:

```
public class SwapInts{

     static int[] myNos=new int[]{56,16,5,12,1960,3,1983,10};

     public static void swap(int[]a,int i, int j){
          int tmp = a[i];
          a[i] = a[j];
          a[j] = tmp;
     }

     public static void main(String[] args) throws Exception{
          swap(myNos, 56, 10);
     }
}
```

The class compiles but when it runs it stops with an exception. The compiler indicates that the source of the error is the main method.

Explain why the class has a run-time error. [4 marks]

(c) Consider the program below:

```
public class FindReplace1{

     static int[] myNos = new int[] {56, 16, 5, 12, 3, 10};

     public static void replace(int[]a, int b, int c){
          int temp = find (a, b);
          if (temp >= 0) a[temp] = c;
     }

     public static int find(int[]a, int x){
          for(int i=0; i<a.length; i++)if (a[i] == x) return i;
          return 0;
     }

     public static void main(String[] args) throws Exception{
          replace(myNos, 99, 404);
          int n = myNos.length;
          for(int i = 0; i<n; i++)System.out.println(myNos[i]);
     }
}
```

Identify the logical error in the above program, and say if you think this will (1) cause a compilation error, or (2) cause a run-time error, or (3) if you think the program will compile and run successfully despite the logical error. [5 marks]

(d) Write a complete Java program called *swapabxy* that takes a file from the command line as a parameter, and writes the contents of the file to standard output, swapping all 'a's for 'x's and all the 'b's for 'y's. For example to run the program on a file called `fred.txt`, we type: `java swapabxy fred.txt`. [8 marks]

Question 5

(a)

(i) What is the output of program *Ex*, below? [4 marks]

```
public class Ex{
     public static void main(String[] args){
          try{
               Integer.parseInt("alpha");
               System.out.println("beta");
          }
          catch(Exception e){
               System.out.println("gamma");
          }
     }
}
```

(ii) What is the output of program *Yex*, below? [4 marks]

```
public class Yex{

     public static void main(String[] args){
          int x = 5; int y = 2;
          try{
               int z = x/y;
               System.out.println("Proceed");
          }
          catch(Exception e){
               System.out.println("divide by zero");
          }
     }
}
```

(b) Consider the Q5 class, below:

```
public class Q5{

     public static boolean Easy(int[ ]a, int thing){
          int size = a.length;
          int first=0,last=size-1;
          int mid;
          while (first <=last){
               mid=first + (last-first)/2;
               if (thing ==a[mid]) return true;
               else if (thing < a[mid]) last = mid-1;
               else first=mid+1;
          }
          return false;
     }

     public static boolean Trex(int[ ]a, int thing){
          int size = a.length;
          if (size == 0) return false;
          if (thing == a[0]) return true;
          int i;
          for (i=0;i<size&&a[i]!=thing;i++);
          return (a[i-1]==thing);
     }
}
```

(i) The two methods in the Q5 class above each implement a well-known algorithm. Can you name the algorithms? [4 marks]

(ii) If you needed to use one of the methods, which one would you choose? Explain your answer. [5 marks]

(c) Write a method with the following heading: [8 marks]

```
public static void insertSpaces(String word)
```

The method takes a `String` and prints it out with two spaces inserted after every character, except the last one. Hence

`insertSpaces("help")` would output `h  e  l  p`

Question 6

(a) Answer true or false to each of the following statements: [8 marks]

(A) Static methods can be run before an instance of the class is made.

(B) Static methods can access instance variables.

(C) Instance variables are normally initialised in a constructor.

(D) Static variables hold the same value for every instance of the class.

(E) Instance methods can be run before an instance of the class is made.

(F) A class can have more than one constructor.

(G) Any method that does not have the keyword *static* in its heading is an instance method.

(H) If *x* is an instance of an Object with its own *toString()* method then `System.out.print(x)` and `System.out.print(x.toString())` will behave in exactly the same way.

(b) Consider the *StringAndInt* class, below:

```
public class StringAndInt{
     public String s;
     public int i;
}
```

(i) Write a constructor for the class. Your constructor should have **two** parameters. [3 marks]

(ii) Write a second constructor for the class. Your second constructor should have **no** parameters, and should initialise the `String` instance variable to "goodbye" and the `int` instance variable to 5 [3 marks]

(iii) Write a suitable *toString()* method for the class. [3 marks]

(c) Consider the *CompareThis* class below:

```
public abstract class CompareThis {
          abstract int compare(int x, int y);
}
```

Write a class *Bigger* that extends *CompareThis* and implements its *compare(int, int)* method. The method should return 1 if *x* is bigger than *y*, -1 if *x* is smaller than *y*, and zero if *x* and *y* are equal. [8 marks]

END OF PAPER