THIS PAPER IS NOT TO BE REMOVED FROM THE EXAMINATION HALLS

UNIVERSITY OF LONDON **CO1109 ZB**

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 *Bool:*

```
public class Bool{

     public static void main(String[] args){
          boolean b = (3==4);
          System.out.println(b);
     }
}
```

Which one of the following is true about class *Bool*: [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 *Bool3:*

```
import java.util.Scanner;

public class Bool3{

     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 *Bool3* 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>6){System.out.print("*");i--;}`

(iii) `for(int i=-2;i<4;i++) System.out.print("*");`

(iv) `int i=3; while (i>23){System.out.print("*");i=i+3;}`

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

(vi) `int i=0; while (i<400){System.out.print("*");i=i+20;}`

(vii) `for(int i=0;i<9;i=i+2) System.out.print("*");`

(viii) `int i=0; while(i<8){System.out.print("*");i--;}`

(ix) `for(int i=4; i>5; i++) System.out.print("**");`

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

```
public class ShapeB{
     public static void main(String[] args){
          for (int i = 3; i >= 0; 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 = 0; i < 4; 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 A {

     public static void main(String[] args){
          int y;
          y = 2.18;
     }
}
```

(ii) [2 marks]

```
public class B{
     String s;

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

(iii) [2 marks]

```
public class C{

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

(iv) [2 marks]

```
public class D{

     static int add(int x, int y){
          int temp = 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("Camelot");`

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

(iii) `int a = Math.abs("Camelot".length())+1966;`

(iv) `Integer.parseInt("350");`

(v) `"carbon".compareTo("350");`

(vi) `"carbon".charAt(350);`

(vii) `int c = "carbon".compareTo("350")+51;`

(viii) `"tread".replace('a',"near".charAt(0)).length();`

(ix) `"tread".replace('a',"near".charAt(0).length());`

(c) Write the following methods:

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

(ii) Write a static method *maxOf3(int, int, int)* which calls *max(int, int)* above.
Your *maxOf3(int, int, int)* method should take three `int` parameters and return the largest of the three. [4 marks]

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 that the object called 'in' is a Scanner object.

(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 `"What is your name?"`.

(B) The program ends with an exception.

(C) The program hangs waiting for the user to enter an integer.

(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 Very {
   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("Dr");
        String w2 = what("Who");
        print("");
        print(w1);
        print(w2);
        print("");
        String[] c1 = dis(1, 1);
        String[] c2 = dis(2, 3);
        String[] c3 = dis(3, 2);
        print(c1);
        print("");
        print(c2);
        print("");
        print(c3);
        print("");
   }
}
```

Consider the *Very* class, above.

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

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

(A) The program will have a run-time error because of a name clash with the two print methods.

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

(C) This is an example of method overriding 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) When does the variable *t* get 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 SwapInt{

     static int[] myNos = new int[] {99, 42, 9, 12, 3, 16};

     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, 99, 16);
     }
}
```

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.

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

(c) Consider the program below:

```
public class FindReplace{

     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);
          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 -1;
     }

     public static void main(String[] args) throws Exception{
          replace(myNos, 99, 9);
          int n = myNos.length;
          for(int i = 0; i<n; i++)System.out.println(myNos[i]);
     }
}
```

The *FindReplace* program has a logical error: with some input given to the *replace(int[], int, int)* method it works as it should and with other input the class will throw a run-time exception. Identify and explain the logical error in the above program. [5 marks]

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

Question 5

(a)

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

```
public class Yez{

     public static void main(String[] args){
          double x = 5.5; double y = 2.0;
          try{
               double z = x/y;
               System.out.println("division done");
          }
          catch(Exception e){
               System.out.println("not a number");
          }
     }
}
```

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

```
public class Ez{
     public static void main(String[] args){
          try{
               Integer.parseInt("delta");
               System.out.println("epsilon");
          }
          catch(Exception e){
               System.out.println("zeta");
          }
     }
}
```

(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) Both methods in the *Q5* class above have been written to perform the same function. Explain what it is that they are intended to do. [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 cannot access instance variables.

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

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

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

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

(F) A class can only have one constructor.

(G) When one class extends another, the keyword *super* can be used in a constructor of the extending class to access a constructor in the class that is being extended.

(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 *IntAndString* class, below:

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

(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 `int` instance variable to 1, and the `String` instance variable to “hello” [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 *Smaller* 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