THIS PAPER IS NOT TO BE REMOVED FROM THE EXAMINATION HALLS

**UNIVERSITY OF LONDON** **CO1109 ZB**

**BSc and Diploma Examination**

**COMPUTING AND INFORMATION SYSTEMS and CREATIVE COMPUTING**

**Introduction to Java and Object-Oriented Programming**

Date and Time: Wednesday 4 May 2016: 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) Java has reference and primitive variables. The names of 4 of the primitive variables types start with the following letters: [4 marks]

i_____

c_____

s_____

d_____

Please name these 4 primitive variable types.

(b) Name an appropriate primitive data type to represent each of the following: [4 marks]

(i) The number of students at a university

(ii) The average age of students at a university

(iii) Whether or not a student has enrolled

(iv) The response of a user asked to type 'y' to continue or 'n' to quit

(c) Write a simple Java program that runs without errors, but does nothing. [2 marks]

(d) Write a simple Java program that prints out the number of command line arguments supplied to it. [3 marks]

(e) Consider the following numbered (from 1 to 13) jumbled up fragments of program code from the class *ToDoList*:

```
1.    if (isDone()) s = s +  "has been done";
           else s = s +"has not been done";
2.  public ToDoList(String item, boolean done) {
3. public class ToDoList {
4.  public static void main(String[] args) {

5.    this.item = item;
      this.done = done;
   }
6.    return s;
         }
7.  public String toString(){
      String s = ("The to do list item: "+item+"\n");
8.  private String item;
    private boolean done;
9.    return done;
   }
10.  System.out.println(toDoList);
11.  }
   }
12.  ToDoList toDoList = new ToDoList("take cat to vet",
           false);
13.  public boolean isDone() {
```

Using **all** the code fragments (and nothing else) write the object-oriented *ToDoList* class. Your completed class should be able to output: [12 marks]

```
The to do list item: take cat to vet
has not been done
```

Your program should follow convention in that the order of appearance of the different elements should be: instance variables; constructor; instance methods; and finally, the main method.

## Question 2

(a) 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. [7 marks]

(i)
```
Integer.parseInt("256");
```

(ii)
```
"dragon".compareTo("unicorn");
```

(iii)
```
"dragon".compareTo(13);
```

(iv)
```
int x = "dragon".compareTo("13")+25;
```

(v)
```
Integer.parseInt(77,8);
```

(vi)
```
Math.abs("Stargate".compareTo("Farscape"));
```

```
FileReader in = new FileReader("Expressions.java");
/* for the next question, assume the above statement
compiles, and that the file "Expressions.java" opens
without error */
```

(vii)
```
Math.max(in.read(),Integer.parseInt("42"));
```

(b) What will be the output of the following statement? [2 marks]

```
"ELI".charAt(("Stargate"+"Farscape").length()/11);
```

(A) `L`
(B) No output - the statement has a compilation error
(C) `I`
(D) No output - the statement will give a run time error
(E) None of the above

(c) 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. [8 marks]

(i)
```
for(int i=0;i<12;i++) System.out.println("*");
```

(ii)
```
for(int i=-1;i<4;i++) System.out.println("*");
```

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

(iv)
```
for(int i=0;i<22;i=i+2) System.out.println("**");
```

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

(vi)
```
for(int i=0; i>5; i++) System.out.println("**");
```

(vii)
```
for(int i=5;i>0;i--); System.out.println("**");
```

(viii)
```
for(int i=10;i>0;i=i-1) System.out.println("*");
```

(d) Write a method whose signature (heading) is [8 marks]

```
static int shortestLength (String [] arr)
```

The method returns the **length** of the shortest `String` in *arr*. It should return zero if *arr* is empty.

**Question 3**

(a) Consider the following classes:

(i) Class *P* has an error. Find it and briefly explain it. [3 marks]

```
class P{
     public static void main(String[] args){
          int s;
          s = "1";
          System.out.println(s);
     }
}
```

(ii) Class *Q* has an error; find it and explain it briefly. [3 marks]

```
public class Q{
     public static void main(String[] args){
          int t = 121 ;
          boolean t = true;
          System.out.println(t);
     }
}
```

(iii) Consider the following three classes, *D, E* and *F*. Two will compile successfully, while one will not. Identify the class that will not compile and briefly explain the error. [3 marks]

```
public class D{

     String z;

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

public class E{
     public static void main(String[] args){
          String z = "hello";
     }
}

public class F{
     static String z;

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

(b)

(i) Rewrite the following code to some other code which behaves the same but which has only one `if`. In your answer assume that the variable *b* is a `boolean` variable, that *t1* and *t2* are both `ints`, and that all three variables have been declared successfully and given a value. [3 marks]

```
if (t1 > t2){
     if (b==true){
          t1=t2+1;
     }
}
System.out.println(t1);
```

(ii) Consider the following code. Simplify it to some other code that behaves the same way. In your answer assume that the variable *y* has been declared as an `int` and successfully given a value, and similarly that the variable *b* has been declared as a `boolean` and successfully given an initial value. HINT: your answer should not have an `else` in it. [3 marks]

```
if(b==true)
     if(!(b==true)) y = y*10;
     else y = 7 + 10;
```

(iii) Rewrite the following method to use a `for` loop instead of a `while` loop [2 marks]

```
public static int CoinToss2(int n){
     int sum = 0;
     int i = 0;
     while (i < n){
          int r = (int)(Math.random()*2);
          sum = sum + r;
          i++;
     }
     return sum;
}
```

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

```
public class ShapeB{

     public static void main(String[] args){

          for (int i = 4; i >= 0; i--){
               for (int j = i; j < 5; j++)
                    System.out.print("*");
               System.out.println();
          }

          for (int i = 0; i < 5; i++){
               for (int j = i; j < 5; j++)
                    System.out.print("*");
               System.out.println();
          }
     }
}
```

## 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 input file that did not exist? [2 marks]

(b) What is the value of the expression `"goodbye".charAt(0)`? [2 marks]

(c) Let *s* be a `String`. [4 marks]

- Write an expression which gives the second character in *s*.
- Write a second expression which gives the last but one character in *s*.

**You may assume *s* has at least two characters.**

(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`. [13 marks]

**Question 5**

(a) Answer true or false to the following statements: [5 marks]

(A) Static methods cannot operate on instance variables

(B) Static methods cannot 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 hold the same value for every instance of the class

(E) Static methods are often used for utility methods

(b) Consider the *MagicEightBall* class below:

```
import java.util.Random;

public class MagicEightBall{

     private static String s1 = "Much is unclear";
     private static String s2 = "It is undoubtedly true";
     private static String s3 = "It is uncertain";
     private static String s4 = "Press forward, not back";
     private static String s5 = "Yes";
     private static String s6 = "No";
     private static String[] answers={s1,s2,s3,s4,s5,s6};
}
```

(i) Write a static method called *eightBall* for the *MagicEightBall* class, that returns a `String` taken at random from the *answers* array. You may wish to make use of the following statements: [2 marks]

```
Random r= new Random();
int randomA = r.nextInt(6);
```

(ii) Write a main method with a test statement for your static method. [3 marks]

(c) Consider the *SixSidedDieToss* class below:

```
import java.util.Random;

public class SixSidedDieToss{

     private static String x1 = "one";
     private static String x2 = "two";
     private static String x3 = "three";
     private static String x4 = "four";
     private static String x5 = "five";
     private static String x6 = "six";
     private String[] tosses={x1,x2,x3,x4,x5,x6};
}
```

(i) Write an instance method called *aToss* for the *SixSidedDieToss* class that will return a random element from the *tosses* array. [3 marks]

(ii) Write a main method for the *SixSidedDieToss* class with test statements to call your instance method. [3 marks]

(d) Write a method with the following signature: [9 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)

(i) What is the output of the following program? [5 marks]

```
class Array1{

     public static void main(String[] args){
       int k=5;
       int num[] = new int[k];
       for (int i=0;i<k;i++)num[i]=i;
       for (int i=0;i<k;i++) System.out.println(num[i]);
     }
}
```

(ii) What happens when this program is run? [5 marks]

```
class f{

     public static void main(String[] args){
          int num[] = new int[5];
          num[5]=2;
     }
}
```

(iii) What is the output of the following program? [4 marks]

```
public class Za{

     public static void main(String[] args){
          try{
               Integer.parseInt("twenty");
               System.out.println("seven");
          }
          catch(Exception e){
               System.out.println("hello");
          }
     }
}
```

(b) Consider the *GolfClubMember* class below:

```
public class GolfClubMember{
  String firstName;
  String familyName;
  double handicap;
  boolean annualFeesPaid;

  public static void main(String[] args){
     GolfClubMember gcm = new GolfClubMember ("Wen", "Ho",
     4.1, false);
  }
}
```

Write a constructor for the *GolfClubMember* class that will make the statement in the main class legal. [4 marks]

(c) Write a method with the signature: [7 marks]

```
static void reverse(String word)
```

The method will reverse the `String` given to it and print the result to standard output; for example `reverse("liar")` gives the output `rail`

**END OF PAPER**