Subject : IB Computer Science HL

Level : Higher Level , year 1

Exam : Final Exam June 2016 - Mr. Mulkey's Class

Date : 2 June 2016 Time: 13:00 - 15:00

Length: 2 hours Contents: 7 pages , 22 questions

Candidate's Name

INSTRUCTIONS TO CANDIDATES:

- **Write your name in the box above.**
- **Do not open the exam paper until told to do so.**
- **Answer ALL questions.**
- **Write ALL answers on the separate answer sheets provided.**
- **Use a black or a dark blue pen to write your answers (no pencils).**
- **Calculators, computers, phones and other digital devices are prohibited.**

****** Possible Printing Errors *******

It is possible that a printing error might occur.
In that case, students should should make a **reasonable** assumption
about what the problem says and continue and answer the question.

Students must assume that ALL Java code is **written correctly** -
there are **no Syntax Errors**. The answer will NEVER be
"the program won't run due to a Syntax Error" – UNLESS the question
clearly states that there is a Syntax Error in the code.

Students must also assume that no Compile Time errors would occur,
UNLESS the problem specifically says that there is a compile-time error.

1. Explain how **ASCII codes** function.
   Then explain the difference between ASCII and UNICODE . *[2 marks]*
   ### ASCII codes are numbers that represent characters,
   for example 'A' ==> 65 ASCII
   Unicode extends ASCII, using 16 bits instead of 8 bits,
   so Unicode represents 65535 characters while ASCII only has 256 characters.

2. State what will be printed by the following commands: *[2 marks]*

```
     String[]  names = {"Alex","Carla","Betty","Doug","Earl"};
     int  count = names.length;
     output( names[ 3 ] );
     output( names[count - 1] );
###  prints   Doug , Earl
```

3. Predict exactly what will be printed by the following commands : *[2 marks]*

```
   int  a = 20;
   int  b = 7 ;
   int  c = a % b;
   int  d = a / b;
   output( c );
   output( d );

###  20 % 7 ==> 6  ,  20 / 7 ==>  2
```

4. (a) Change the binary number **01010011 bin** to the equivalent **decimal** value (base 10). *[1 mark]*

   ### 64+16+2+1 ==> 85

   (b) Change the decimal number **123 dec** to the equivalent **binary** number. *[2 marks]*

   ### 123 = 64 + 32 + 16 + 8 + 2 + 1 ==> 01111011

5. A Laptop computer is likely to contain both RAM memory and Hard Disk storage.
   For each of these 2 types of storage, describe one function that it
   can perform that would not be performed by the other, and include
   an explanation of why that function is best performed by that memory type. *[4 marks]*

   ### RAM contains active (running) programs and their data.
   HD contains permanent copies of program.
   RAM is not permanent, gets erased when the power is turned off, so cannot permanently store programs.
   RAM is much faster than HD - it is not possible to run programs directly from disk -
   the programs must first be copied into RAM before running.

6. It is possible to store a NUMBER in a String variable, like this:

   String value = "135" ;

   Explain why it is usually a bad idea to store numbers in String variables. *[3 marks]*

   ### The computer cannot do arithmetic operations on String values.
   For example, "123" + "456" ==> 123456 (concatenation, not addition)

7. Describe a situation where a user might type a number and it would cause an OVERFLOW error. Explain why the error occurs. *[2 marks]*

### You cannot type in a number larger than 2 billion if it is stored in an int variable.
For example, typing in 60 billion would cause an overflow if stored in an int variable.
The int variable only uses 32 bits for storage - the max value is 2^31-1 == 2 billion

8. Explain how classes and objects are related, and also how they are different. *[2 marks]*

### a class is a description of a collection of data and methods, for example:

```
public class Student
{
    String name;
    int age;

    public void birthday()
    {    age = age + 1; }
}
```

The class is a "blueprint" (design) for creating OBJECTS.

```
For example:   Student boy = new Student();
               Student girl = new Student();
```

creates 2 student objects, and each has its own name and age.

9. It is possible to input a DATE as a String, like "4 July, 1776", and then store it in a Date type variable. That converts the String into an equivalent number, as the number of milliseconds since 1 Jan 1970.

   Describe one advantage of storing a date as a number of milliseconds rather than storing it as a String. *[2 marks]*

### With the milliseconds version it is possible to do calculations,
like calculating the number of days between end of school (25 June) and start of scholol (15 Aug).
There is no easy way to count these days using the String versions.
But with milliseconds, you subtract and divide by the number of milliseconds in a day.

10. Explain what "parallel arrays" are, including a specific example of when they are needed instead of one single array. *[2 marks]*

### Parallel arrays are two (or more) separate arrays, each containing a different data item.
For example : names[ ] and ages[ ]
Each item in names corresponds to the "parallel" item in ages, both for one person.

# 11. Assume that the array **PRICES[ ]** contains exactly **100 double** values, e.g. {12.99 , 2.50 , 29.00 , ...}

(a) Write a method called **countSmall( )** that counts how many of the values are below 10.00 *[4 mare*

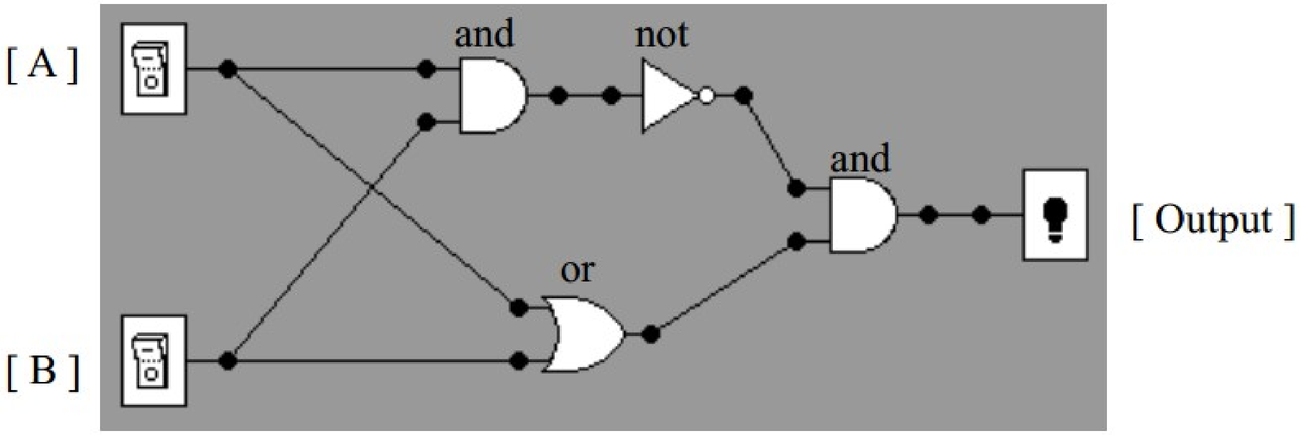
### Be sure to INITIALIZE the count variable to 0 BEFORE starting the loop.

(b) Write a method that will **sort** the numbers in order, from largest to smallest. *[4 marks]*

### Do NOT use the SimpleSort method that the teacher showed you (and told you never to use).
You must write a standard algorithm - BubbleSort or SelectionSort or InsertionSort.

====================

# 12. **About Logic Gates**



(a) Study the circuit above. Then fill in the following truth table for the circuit: *[4 marks]*

A	B	Output
0	0	
0	1	
1	0	
1	1	

### 0 , 1 , 1 , 0 -- you should copy the ENTIRE table onto your answer sheet.

(b) Explain the difference between **Inclusive OR** and **Exclusive OR (XOR)** gates. *[2 marks]*

### Inclusive OR	Exclusive XOR	
0	0	*Inclusive or = Either, OR, or both*
1	1	*Exclusive = Either, OR, but NOT both*
1	1	
1	0	

(c) Explain why **NAND gates** are commonly used in USB Stick storage chips. *[2 marks]*

### Using NAND gates, it is possible to constructor all other gate behaviors, so others aren't needed.
And NAND gates are simpler and use less power than other types of gates.

#13

(a) Outline 2 differences between a **Linked-List** and an **Array.** *[2 marks]*

### Both are linear lists of data.
The capacity (.length) of an array must be specified at the beginning of the program, and this .length cannot change during the program - the array length is STATIC.
For example, String[] names = new String[100]; can store 100 String values.
A LinkedList is dynamic, and can grow or shrink (change size) during the program.
In an array, new values must be written to a specific cell, replacing previous data.
In a LinkedList, it is possible to insert (.add) a new item in between two other items.

(b) Explain the purpose of the method: **LinkedList.remove(int p);** *[2 marks]*

### This removes an item, specifically the item that is at POSITION P.

# 14. The following PSEUDOCODE algorithm adds up the numbers 1 + 0.5 + 0.25 + 0.125

```
SUM = 0
X = 1
SUM = SUM + X
X = X*0.5
SUM = SUM + X
X = X*0.5
SUM = SUM + X
X = X*0.5
SUM = SUM + X
```

(a) Assume the program will be changed to add up 1 + 0.5 + 0.25 + 0.125 + ... + (0.5)^999999
or stated differently: 0.5 ^ 0 + 0.5 ^ 1 + 0.5 ^ 2 + ... + 0.5 ^ 999999 (1 million terms).
Outline **two reasons** that a **loop** should be used, rather than just writing a lot more commands. *[2 marks]*

### (1) it would be a LOT of work to copy and paste 1 million copies, and probably be incorrect.
(2) with a loop, it is easy to CHANGE the loop to add up a different number of items

(b) Write **Java commands** using a **loop** to add up: 0.5^0 + 0.5^1 + 0.5^2 + ... + 0.5^999999 *[4 marks]*

```
 double  sum = 0.0;
 double  val =  1;
 for(int  x = 0; x < 1000000; x=x+1)
{     sum = sum + val;
       val = val * 0.5;
}
output(sum);
```

==

# 15. Assume that 10 international schools in 5 different countries wish to form a partnership. They will share information about courses that they teach, as well as personal information about students and teachers. The new INTSCHOOLS partnership will need computer systems.

(a) Outline 2 advantages of allowing each school to use their own favorite computer systems. *[2 marks]*

### Each school probably has an existing infrastructure of computers, networks and software. If they continue to use these systems, they do not need to purchase new systems, and less new training will be required for the users.

(b) Outline 2 disadvantages of allowing various schools to choose different computer systems. *[2 marks]*

### This would require the new computer system(s) be be compatible with many different systems. This decreases reliability and increases support problems for technicians.

(c) Assume that INTSCHOOLS decides to purchase a large computer system that will be used in the same way in all the schools. *[2 marks]*

Outline 2 reasons that there will be problems connected with having schools in several countries all using the same computer systems.

### Different time zones, calendars and languages cause compatibility issues. This may cause considerable training and usability issues in various locations, since the users are currently using a variety of different systems.

==

# 16. The following questions are about **Operating Systems** in a laptop (notebook) computer such as a PC laptop or a MacBook.

(a) Explain why Operating Systems must include **memory management** features, including how this is related to loading programs from the hard-disk. *[2 marks]*

### When a software application starts, the OS must copy it from the Hard Disk and put a copy into the RAM memory. Since many programs can be in the memory at the same time, the OS must decide WHERE in the memory to put the program, so that it does not OVERWRITE another program. That means the OS must manage the locations of programs and keep track of where the FREE MEMORY is located. Furthermore, it is possible that all the running programs will completely fill up RAM. Then the OS can SWAP some programs onto the disk drive (in virtual memory) to create some free space for a new program to be loaded.

(b) Explain how **multi-tasking** works, including a specific example of when it is beneficial to a user. *[2 marks]*

### Multi-tasking means that many programs can be loaded into RAM simultaneously. They all appear to be running at the same time. In actual fact, the CPU is switching between tasks frequently and quickly, running each task for perhaps a few milliseconds before switching to another task. For the user, it appears than all these tasks are continuing to execute simultaneously, so a user can watch a video while typing an essay (like I am do right now).

17. Assume that the class EMAIL (below) is used to store eMail addresses and names.
Each EMAIL object will contain the eMail **address** and **name** of a user.

```
public class EMAIL
{
   String  address = "";
   String  name = "";
}
```

The **collect( )** method below inputs eMail addresses and stores them in LinkedList DATA .
After running **collect( ),** DATA contains lots of EMAIL objects. This method is part of a larger program.

```
LinkedList<EMAIL>  DATA  = new LinkedList<EMAIL>();

public void collect()
{
   String MORE = "Y";

   while( MORE.equals("Y") )
   {
       String  N = input("Name ?");
       String  A = input("eMail address?");
       EMAIL temp = new EMAIL();
       temp.name = N;
       temp.address = A;
       DATA.add(temp);

       MORE = input("Type another name ? (Y/N)");
   }
}
```

After running the **collect( )** method, the **printAll( )** method (below) will print all the eMail addresses.

```
public void printAll()
{
   int count = DATA.size();

   for(int x = 0; x < count ; x=x+1)
   {
       EMAIL info = DATA.get(x);
       System.out.println( info.address );
   }
}
```

(a) Write a Java method called **findName(String find)** . This method accepts **find** as a parameter. The method must search in DATA for a **name** that matches **find** . Whenever it finds a matching name, it should print out the email **address** for that person. If there are many addresses for the same name, it should print all of them. *[4 marks]*

(b) Write a new method called **checkEmail(String address)** . This method accepts an eMail address as a parameter. It checks the following: (1) **address** contains the '@' symbol ; (2) there are at least 2 characters to the left of the '@' symbol ; (3) the **address** contains a period '.' symbol ; (4) the period '.' symbol occurs to the right of the '@' symbol. If any of these 4 checks fails, then the method returns "bad". Otherwise, the method returns "good". *[6 marks]*

#17 solutions:

```
public void  findName(String find)
{
   int max = DATA.size();
   for(int x = 0; x < max ; x = x+1)
   {
      String name = DATA.get(x).name;
      if(name.equals(find))
      {
         output( DATA.get(x).address );
      }
   }
}

public String checkEmail(String address)
{
   int at = address.indexOf("@");
   int dot = address.indesOf(".");
   if( at >= 2  &&  dot > at)
   {  return "good"; }
   else
   {  return "bad"; }
}
```

===== Questions 18-22 are for HIGHER LEVEL STUDENTS ONLY ================

#18 A clothes washing machine probably contains a CPU (microprocessor).
Outline 2 reasons that the washing machine does NOT need a GUI operating system. *[2 marks]*

### *A GUI OS provides 2 important features:*

(1) a very flexible sequence of actions, responding to the sequence of user inputs (events)
(2) a very flexible variety of graphical outputs, especially pictures

Washing machines have a very limited set of "programs", for various types of laundry. These programs are not flexible or changeable. So it's not necessary to respond to a variety of orders of user actions. Also, the the outputs from the washing machine are quite simple and there would be little benefit from displaying a large variety of outputs and pictures.

#19 Explain the major difference between a STACK and a QUEUE. *[2 marks]*

### *Stack = Last In First Out (LIFO), like the passengers in a narrow elevator or a bus*

Queue = First In First Out (FIFO), like the customers in a line in a cafeteria

----

#20 An MP3 music player might use **encryption** or **compression**.
Decide which is more useful in an MP3 music player
and explain how it is used in the music player. *[2 marks]*

### *Encryption scrambles information so that only someone with a password can read or understand it. There is no need for secrecy in an MP3 player. On the other hand, compression makes files smaller - for example, an MP3 audio file is typically 10 times smaller. So, in an MP3 player with limited memory (for example 8 GB), it is possible to store lots more songs if they are compressed. The compression happens BEFORE copying the files into the MP3 player.*

Then, when a song is played, the MP3 player must decompress the file in real time, before sending the music to the headphones.

#21 - This question is about **ROBOTS**

(a) Assume that a vacuum cleaning robot moves around in a room and sweeps up dirt.
Explain the roll of **sensors** in this robot, and explain why it would
not be a good idea to create such a robot WITHOUT sensors. *[3 marks]*

### *As the vacuum cleaner moves around the room, it will encounter obstacles like walls and chairs. Sensors are used to detect obstacles. The sensors might receive optical or radio (RADAR) signals. When a sensor detects an obstacle, the processor and software in the vacuum must decide which way the robot should turn to avoid the obstacle. Without sensors, the robot would either run into objects and stop, or more likely it would need to be programmed ahead of time to follow a specific path. That might work in a relatively empty room, and in a room where nothing ever moves (e.g. chairs), but would require considerable effort to input the program ahead of time. It's much easier to allow the robot to decide AUTONOMOUSLY how to avoid the obstacles.*

(b) Explain what the term **AUTONOMOUS** means, and why this is significant for robots. *[2 marks]*

### *Autotonomous means that a robot can make decisions that control its behavior without requiring direct control from a user. This allows a robot to carry on doing work when there is no user available to make decisions and provide control.*

(c) Explain why a robot would probably not be using a Windows or Mac operating system. *[3 marks]*

### *Windows and Mac OS contain lots and lots of features that are of no value for a robot. For example, a robot probably does not need a GUI interface, and does not need to load lots of programs from a hard-disk. Since it does not need these features, there is little point installing a large OS that requires significant amounts of RAM and Hard-Disk Storage in order to function. Cost may also be an issue, where the cost of a desktop OS is on the order of 50 Euros. That wouldn't matter in a very large, expensive industrial robot, but might be too expensive for a small, cheap vacuum clearner.*

=====================================

#22 Assume that a **two-dimensional array** named **MONEY** contains **4 rows** and **20 columns**. Part of the MONEY array data is shown here.

14	35	25	18	27	9	.......more columns here
10	20	30	16	21	4	.......more columns here
15	46	17	14	13	1	.......more columns here
0	0	0	0	0	0	.......more columns here

//-- Study the code below. --//

```
int num = 0;

for(int row = 0; row < 4; row = row + 1)
{
   num = num + MONEY[row][3];
}

output( num );
```

```
//------- end of code ----------//
```

(a) Explain what will be printed by the code above. *[2 marks]*

### *This adds up the numbers in column #3 = 18+16+14 = 48*

(b) Write a Java method that USES LOOPS to :

- add up the total of each COLUMN, and save the total in the bottom row, where the 0 is written
  This produces 20 results, one for each column.

- then examine all the results in the bottom row,
  and find the LARGEST TOTAL and output this result. *[8 marks]*

```
void findLargestTotal()
{
   for(int col = 0; col < 20; col = col+1)
   {
      int sum = 0;
      for(int row = 0; row < 3; row = row + 1)
      {
         sum = sum + MONEY[row][col];
      }
      MONEY[3][col] = sum;
   }

   int max = MONEY[3][0];
   for(int col = 0; col < 20; col = col+1)
   {
      if( MONEY[3][col] > max )
      {
          max = MONEY[3][col];
      }
   }
   output("Largest = " + max);

}
```