May 2015 – Computing Advanced
Paper 1

Question 1

a. It will output "It's NOT Rover"
b. Class Main should be changed to the following (bold characters show the changes)

```
public class Main
{
   public static void main (String[] args)
   {
      Dog aDog = new Dog ("Rover");

      aDog = changeDog (aDog);

      if (aDog.getName().equals("Rover"))
        {
           System.out.println ("It's Rover");
        }
        else
        {
           System.out.println ("It's NOT Rover!");
        }
   }

   public static Dog changeDog (Dog d)
   {
      d.setName ("Fifi");
      return d;
   }
}
```

Question 2

a. In object-oriented languages (OOLs) data and instructions are grouped together (encapsulation). In traditional imperative languages (ILs) encapsulation is not present.
b. In OOLs a program is divided in a number of classes. In ILs a program is divided in a number of procedures (or functions). In OOLs this permits data hiding since some variables and methods can be declared as 'private'.
c. OOLs have inheritance. ILs do not have it.
d. OOLs possess the feature of polymorphism (overloading and overriding). ILs do not have this feature.

e. OOLs have data hiding (when variables and methods are private). This is not present in ILs.

Question 3

a. $0F_{16} = 00001111_2 = 15_{10}$
b. 01111000
c. $01111000_2 = 120_{10}$
d. Multiplication by 8

Question 4

AB'C' + A'BC + AB + AC + B'C	
= B'(AC' + C) + A'BC + AB + AC	(distributive law)
= B'(A + C) + A'BC + AB + AC	(using the law X + X'Y = X + Y)
= B'(A + C) + B(A'C + A) + AC	(distributive law)
= B'(A + C) + B(A + C) + AC	(using the law X + X'Y = X + Y)
= (B' + B)(A + C) + AC	(distributive law)
= 1.(A + C) + AC	(tautology law)
= A + C + AC	(tautology law)
= A + C	(law of absorption)

Question 5

a. Size of address bus = $\log_2 256K = \log_2 256x2^{10} = \log_2 2^8x2^{10} = \log_2 2^{18} = 18$
b. 18

Question 6

a. When a function is called (by means of the CALL assembly instruction), the address of the next program-line is pushed onto the stack so that when the function ends its execution, the program can pop the top of the stack and know where to proceed with the next instruction.
b. When the RET assembly instruction is met, the stack is popped so that the program can proceed from the popped address.

Question 7

	Instruction	Number of instructions	Value of CX
step 1	MOV CX, 3	1	CX = 3
step 2	CALL FN	10	

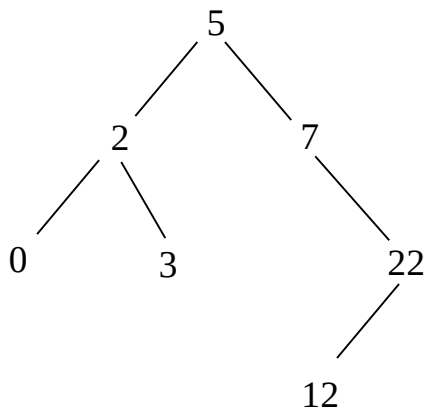
step 3	DEC CX	1	CX = 2
step 4	JNZ LOOP	1	
step 5	CALL FN	10	
step 6	DEC CX	1	CX = 1
step 7	JNZ LOOP	1	
step 8	CALL FN	10	
step 9	DEC CX	1	CX = 0
step 10	JNZ LOOP	1	
		37	Total number of instructions

Question 8

a. You use an assembler when you have a program written in assembly language and you want to translate it into executable code.
b. You use a translator when you want to produce executable code from non-executable code.
c. You use a compiler when you have a program in a high-level language and you want to convert it to an executable program (i.e. in machine code).
d. You use an interpreter when you want to execute a program in a high-level language perhaps when it is not all debugged.
e. You use a cross compiler when you want to translate code into code of a different machine than the one executing the compiler. The cross compiler can be used for example when the computer for which the translation is produced has a very small memory e.g. an embedded computer within a washing machine.

Question 9

a. A binary tree is a tree whereby each node has not more than two children.
b.



Question 10

A terminal symbol is a symbol that makes part of the vocabulary of a language. Non-terminal symbols are meta-language symbols that are not part of the vocabulary of a language. In the following example <digit> and <integer> are non-terminal symbols while all the others are terminal symbols.

<digit> ::= '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'
<integer> ::= ['-'] <digit> {<digit>}

Question 11

a. Any two from: lexical analysis, syntax analysis, semantic analysis, intermediate code generation, code optimization, code generation.
b. One of
    i. Lexical Analysis: This phase scans the source code as a stream of characters and converts it into meaningful lexemes. The lexical analyser represents these lexemes in the form of tokens.
    ii. Syntax Analysis (parsing): It takes the token produced by lexical analysis as input and generates a parse tree (or syntax tree). In this phase, token arrangements are checked against the source code grammar, i.e. the parser checks if the expression made by the tokens is syntactically correct.
    iii. Semantic Analysis: Semantic analysis checks whether the parse tree constructed follows the rules of language for example, it checks if assignment of values is between compatible data types. Also, the semantic analyser keeps track of identifiers, their types and expressions; checks whether identifiers are declared before use or not etc. The semantic analyser produces an annotated syntax tree as an output.
    iv. Intermediate Code Generation: After semantic analysis the compiler generates an intermediate code of the source code for the target machine. It represents a program for some abstract machine. It is in between the high-level language and the machine language. This intermediate code should be generated in such a way that it makes it easier to be translated into the target machine code.
    v. Code Optimization: The next phase does code optimization of the intermediate code. Optimization can be assumed as something that removes unnecessary code lines, and arranges the sequence of statements in order to speed up the program execution without wasting resources (CPU, memory).
    vi. Code Generation: In this phase, the code generator takes the optimized representation of the intermediate code and maps it to the target machine language. The code generator translates the intermediate code into a sequence of (generally) relocatable machine code.

Question 12

a. This incorporates all programs that are required to run the computer. A few examples are operating system, antivirus and compiler.
b. One of the following:
    i. Waterfall model: each phase must be completed fully before the next phase can begin. It is basically used for small projects where there are no uncertain requirements. In this model the testing starts only after the development is complete. One advantage is that it is easy to manage because it is very clear what each phase has to deliver. One disadvantage is that it is very difficult to go back and change something. This model is used only when the requirements are very well known, clear and fixed.
    ii. Spiral model: This model has four phases which are (1) planning, (2) risk analysis, (3) engineering and (4) evaluation. The project repeatedly passes through these phases in iterations (called Spirals in this model). In the planning phase asks the question: what is required of the system? In the risk analysis phase a process is undertaken to identify risk and solutions and a prototype is produced. At the engineering phase software is developed. In the evaluation phase the customer evaluates the output of the project before possibly going to the next spiral.
    iii. RAD model is Rapid Application Development model. Components or functions are developed in parallel as if they were mini projects. The developments are time boxed, delivered and then assembled into a working prototype. This can quickly give the customer something to see and use and to provide feedback regarding the delivery and their requirements. Application generators are also used. Two advantages of RAD are that it reduces development time and encourages customer feedback.
    iv. Prototyping: The basic idea here is to build a rudimentary working model so that the customer can participate in shaping the final product. Some types of prototyping are the following: (1) throwaway prototyping where the initial model is only used for demonstration purposes, (2) evolutionary prototyping improves on the initial prototype until the project is ready, (3) incremental prototyping means that the final product is built as separate prototypes.

Question 13

a. Design stage
b.
    i. JSP is a method for program design. It is based on the decomposition of a problem in smaller problems. The diagram is hierarchical. It has ways to show sequences, selections, iterations, subroutine calls and parameters.

ii. Flowcharts are visual representations of the logic of a program. Flowcharts use symbols by means of which a program's logic can be shown in detail.

Question 14

- No unnecessary data duplication (this leads to more efficient storage).
- No data inconsistency.
- Simpler to delete or modify details.
- Complex queries can be carried out (by means of SQL)
- Better security – different users can be given different access to different tables.
- The database can be modified relatively easily.

Question 15

a.

first_name	last_name
Wendy	Bertuzzi

b.

employee_id	first_name	salary
3	Lionel	50000.00
4	Alan	25000.00
5	Wendy	15000.00
8	Helen	83000.00

Question 16

a. Four of the following: batch, multiprogramming, timesharing, multitasking, distributed, network, real-time, single-user, multi-user, multiprocessing, multithreading, online and embedded operating systems.
b. One of the following:
   i. Batch operating system: The programs are executed in sequence and thus it cannot run interactive programs. JCL (Job Control Language) is a command language used with the batch OS. It specifies, for example, priority, program size as well as the files and databases used.
   ii. Multiprogramming Operating System: This is an improvement on the batch OS. In the processor's wait states another process is executed.
   iii. Time-sharing operating system: It enables many users to use the same computer at the same time. The processing of the programs seems concurrent however the CPU only processes each program individually for a quantum of time before passing one by one to the

other programs and then repeats the process forever. For some it is identical to multitasking, for others it is synonymous to multiuser.

iv. Multitasking: Multitasking is like time-sharing but it acts only on one processor. Multitasking can be pre-emptive or cooperative.

v. Distributed operating system: It makes use of resources found on a number of linked computers like processors and RAM. Processors in a distributed system may vary in size and function so the OS dispatches jobs to the most appropriate processor.

vi. Network operating system: it manages applications on the server, data and users e.g. checks the users logging on. It manages security.

vii. Real Time operating system (RTOS): the operating system is bound with a time interval to give an output. There are two types of RTOSs: hard (critical) and soft (non-critical). In the hard RTOS response from the system cannot have any delays while soft systems are less restrictive.

viii. Single-user operating system: it can have only a single application running (e.g. on a mobile phone) or it can allow many applications to be open (e.g. a desktop with multitasking properties.

ix. Multi-user operating system: This is an operating system that supports two or more simultaneous users e.g. on mainframes or minicomputers. This is synonymous with timesharing.

x. Multiprocessing operating system: This is time-sharing on multiple CPUs. Multiprocessing is divided into (1) Symmetric multiprocessing (SMP) (where the processors share memory and the I/O bus or data path and (2) Massively parallel processing (MPP) where each processor has its own operating system and memory.

xi. Multithreading operating system: It has the structure to execute threads in parallel taking care of the usual memory management, process management, security etc. Threads can belong to an application or to the operating system.

xii. Online operating system: This operating system downloads applications, checks who the connecting users are, on your computer is responsible for the installation and removal of applications. It enables you to use the programs by downloading the application modules into memory and by uploading requests and necessary all the necessary files to the sender.

xiii. Embedded operating systems: These are usually used for hardware that have very little computing power, little RAM/ROM and a slow CPU, so they tend to be very specific in their applications and scope. Embedded OSs can be found in cars, large laser printers, some home appliances, and even military systems.

Question 17

a. Processes need management because of a number of reasons e.g. critical sections cannot be executed at the same time (this would potentially cause erratic results); processes can enter a deadlock; processes can require simultaneously the same resource etc.

b. (i) Executing: when the process is being executed by the CPU; (ii) ready (waiting): when the process is waiting to be dispatched to the CPU to be executed; (iii) blocked (suspended): when the process cannot proceed because it is waiting for an event to happen e.g. an input.

Question 18

a. This is one way to organize memory to hold programs being executed. RAM is divided into a number of partitions and each partition will hold a program in it.
b. Three other ways to do the same thing are the following:
    i. Dynamic partitioning with compaction (when a program wants space in RAM the OS finds a free 'hole' which is large enough to hold the program; every so often compaction is applied to eliminate (momentarily) fragmentation).
    ii. Paging (RAM is divided into pages and a program is assigned a number of pages; only parts of programs are kept in the RAM and if a part of a program is required this is put in a page that will be cleared from its contents which will be copied on disk).
    iii. Segmentation (a program is divided into segments; only the required segments are put in the RAM; a reference to a location in a segment consists of (i) the number of the segment, (ii) the address within the segment).

Question 19

a. It is a number used to indicate the location of a computer or other device on a network using TCP/IP.
b. An Internet application is a program that you use which is not installed on your computer but which you can access over the Internet. Examples include:
    i. Google Apps: it includes Google Docs (a word processor), Google Sheets (a spreadsheet) and Google Slides (a presentation program).
    ii. Picture or video applications where pictures or videos can be edited with programs located on a remote server.
    iii. Online streaming video applications.

Question 20

a. Transmission error.
b. Checksum, CRC, parity check.
