

**THE AUSTRALIAN NATIONAL UNIVERSITY**

*First Semester Examinations 2005*

**COMP1200**  
**(Perspectives on Computing)**

*Writing Period: 3 hours duration*  
*Study Period: 15 minutes duration*  
*Maximum Marks: 100*  
*Permitted Materials: None*  
*Answer all questions.*

Name
Student Number

PLEASE READ THE FOLLOWING INSTRUCTIONS CAREFULLY.

- This paper will be marked out of 100 and consists of 6 questions. Questions are of unequal value. The value of each question is shown within square brackets. Questions that are partitioned into parts show the number of marks given to each part within square brackets. Students should attempt all questions.
- Answer *all* questions using either a **black** or **blue** pen. Use the space provided. Marks may be lost for giving information that is irrelevant. There is additional space at the end of the booklet in case the space provided is insufficient. If you do need to use this extension space, clearly indicate in the first part of your answer that the answer is continued at the end.
- Students are permitted to have pens, pencils, rulers, and erasers. However, no other materials are permitted.
- Students are asked to check that this examination paper contains all 18 pages.
- This examination paper is **CONFIDENTIAL** and is not to be taken from the examination room. No pages are to be torn from this examination paper.

Official use only:

1	2	3	4	5	6	Total
---	---	---	---	---	---	-------

1. [16 Marks] Computer Architecture

- a) [4/16] Referring to measurements of amounts of computer storage, state which is the *largest* and which is the *smallest* among the following:

4096 K bits	2 MBytes	$2^{10}$ bits
2000 Kbits	2048 Bytes	100,000 Bytes
1000 bits	3125 Kbits	101,110 Bytes

The largest is \_\_\_\_\_ . The smallest is \_\_\_\_\_ .

- b) [2/16] The equivalent of the binary number 10110100 in hexadecimal notation is

\_\_\_\_\_

- c) [2/16] If this same bit pattern 10110100 is regarded as an 8-bit, two's complement number, it represents the decimal number \_\_\_\_\_  
*show your working steps here*

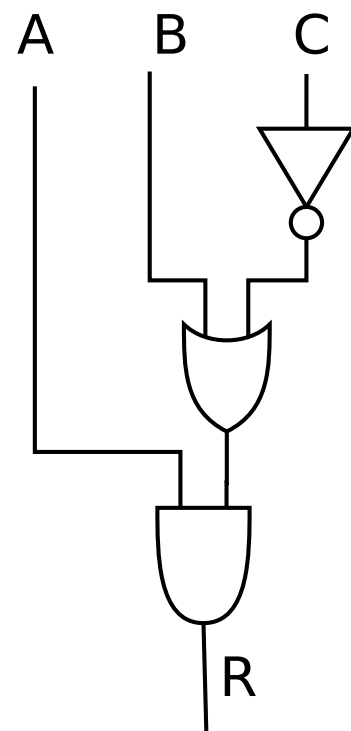
- d) [2/16] The representation of the decimal number +18 as an 8 bit, two's complement

binary number is \_\_\_\_\_  
*show your working steps here*

- e) [3/16] Explain briefly how the behaviour of flip-flops is significantly different in nature from the other kinds of computer logical circuits that calculate simple logical functions, such as a binary adder.

- f) [3/16] Complete the logic table for this logic circuit.

A	B	C	R
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	



QUESTION 2. [18 marks] Operating Systems and Networks

- a) [5/18] Using a diagram, describe the steps involved when the operating system causes a Context Switch from one process (called A) to another process (called B), using Process Control Blocks.

For parts (b), (c) and (d) of this question, consider a simple multiprocessing operating system in which processes can be in one of five states: *Running*, *Waiting*, *Ready*, *New*, or *Terminated*.

This operating system is executing on a simple computer which has only one, simple CPU. The operating system allows a large number of processes to exist.

- b) [2/18] How many of the processes can be in **Running** state *at one time*?

Answer 0, 1, 2, 3, or "many" (more than 3): \_\_\_\_\_

- c) [2/18] How many of the processes can be in **Waiting** state *at one time*?

Answer 0, 1, 2, 3, or "many" (more than 3): \_\_\_\_\_

- d) [4/18] What is the state of the two processes A and B referred to in question part (a), before and after the context switch takes place?

	<i>process state</i>	
	before	after
A		
B		

- e) [5/18] Briefly explain how an operating system that allows context switching can make it possible in general for the computer to finish a given mixture of tasks faster compared to a system that does not allow context switching between processes.

QUESTION 3. [20 marks] Computation

a) [6/20] Fill in the blanks in the following sentences:

- Two examples of the vector graphics approach of representing images are \_\_\_\_\_ and \_\_\_\_\_ .
- The first two stages of the compilation process are (1) \_\_\_\_\_ and (2) \_\_\_\_\_ .
- Two of the key terms used in a proof of correctness are \_\_\_\_\_ and \_\_\_\_\_ .

b) [2/20] Differentiate between *lossy* and *lossless* image compression.

c) [3/20] Consider a program that maintains a car's speed at a preset value. Would you consider such a program to be an algorithm? Briefly explain.

d) [4/20] Consider the following program, where  $n$  is a positive integer.

```
i := 1; j := 0;

while (i < n)
do
  j := (i * i);
  j := i + (j * j);
  i := i + 1;
end
print j;
```

- 1) What is the running time of this program, using the  $O$  notation, and why? Show your working. Assume that each of the following instructions takes unit time: addition, subtraction, multiplication, comparison, assignment and print. [2]
- 2) Give the output of the program when  $n = 5$ . [1]
- 3) Suppose that you write another program that performs the same computation as above, but runs in  $O(n \log n)$  time. Which program, in general, runs faster, and why? [1]

e) [2/20] What is the relationship between Turing machines and algorithms?

f) [3/20] Briefly describe the *declarative* programming paradigm. Give an example of a declarative programming language.

QUESTION 4. [20 Marks] The History of Computing

A list of significant dates taken from the course web-notes is included at the end of the question paper.

Consider a typical large mainframe computer built around 1970.

a) [2/20] What generation of computer hardware is this mainframe computer?

\_\_\_\_\_

b) [2/20] What kind of electronic circuitry would be used for its CPU?

\_\_\_\_\_

c) [2/20] What is the highest generation of computer programming languages that is likely to be used on this computer? (*answer 1, 2, 3, 4, ...*) \_\_\_\_\_

d) [2/20] Name one language that is likely to have been used on this computer for programming *scientific* problems \_\_\_\_\_

e) [2/20] Name one language that is likely to have been used on this computer for programming *commercial* problems \_\_\_\_\_

f) [4/20] State Moore's Law, including the value that is now generally accepted as the "doubling time" in Moore's Law.

\_\_\_\_\_  
\_\_\_\_\_

g) [2/20] Name two other properties of computer processor chips that have shown an exponential improvement similar to the property described in Moore's Law.

a) \_\_\_\_\_

b) \_\_\_\_\_

h) [4/20] A modern computer designed since 2000 has many more transistors in its processor than did the computer of 1970. State 2 ways in which the modern CPU design uses these extra transistors to improve the speed of the processor:

a) \_\_\_\_\_

b) \_\_\_\_\_

QUESTION 5. [17 Marks] Software Engineering

- a) [5/17] With the aid of a diagram, describe the Waterfall Model of software development. You should mention the role of feedback arrows (if any) in the diagram.

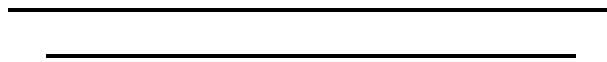
- b) [2/17] Contrast the technical terms verification and validation.
- c) [2/17] In what circumstances might a lawyer be looking at an SRS (Software Requirements Specification).
- d) [3/17] Problems found by a consumer in a software product can sometimes be attributed to an error caused by ambiguity in an English description.  
In what phase of the lifecycle is this most likely to occur? Why are such problems particularly expensive to correct?
- e) [5/17] Three friends, X, Y and Z, come to university to do computing. X does a computer science degree, Y does a BIT in information systems and Z does a BSEng. After four years and about the same number of computing related courses they graduate and form a company to develop software. Contrast the sort of roles they are best equipped to take on.

QUESTION 6. [9 Marks] HTML and AI

a) [3/9] Using examples, briefly describe the three different types of HTML links.

b) [3/9] Briefly explain the difference between *inductive* and *deductive* reasoning. What form of reasoning does *learning* use?

c) [3/9] Describe how the problem of travelling from one city to another could be framed as a production system. What are the states? What are the productions?



Additional answers to question number: \_\_\_\_.

Additional answers to question number: \_\_\_\_.

Additional answers to question number: \_\_\_\_.

## A brief list of dates for Perspectives on Computing

- 1935 IBM 601 punched card machine using relays (USA)  
1937 Harvard Mark I IBM ASCC: 72 x 23 decimal digit number storage: multiply time 6 seconds  
1938 Konrad Zuse: electro-mechanical computer Z1 (DE)  
1939 Atanasoff and Berry: special purpose electronic computer (USA)  
1941 Atanasoff and Berry: linear equation solver (USA)  
1941 Konrad Zuse: Z3 programmable computer (Germany): telephone relays with mechanical storage for 64 numbers  
1943 Harvard Mark I programmable computer (USA)  
1943 Thomas Flowers: Colossus - the first electronic computer (UK)- 2000 vacuum tubes
- 1946 ENIAC fully electronic computer (USA) Eckert and Mauchly: 20 x 10-digit accumulators; 5000 operations per second  
1946 Konrad Zuse's programming language Plankalkül  
1946 von Neumann's paper on computer architecture: "the Von Neumann computer"  
1946 Grace Hopper logs the first computer "bug" (a term previously used widely in engineering)  
1947 transistor invented (Bell Labs USA)  
1948 First stored-program computer (UK)  
1949 CSIRAC: the first Australian-designed and built computer  
1949 First use of magnetic tape (USA)
- 1950 Floppy disk invented (Japan)  
1951 First commercially available computer (USA)  
1951 symbolic assembler language invented (Grace Hopper) to improve on programming by numeric codes  
1951-55 IBM 701 (scientific) and IBM 702 (commercial): 50 of each were sold  
1953 Magnetic core memory  
1956 television introduced in Australia  
1957 (approx)languages FORTRAN(1954-7); Algol (1958-60); COBOL (1959-61); LISP (1956-59)  
1958 FORTRAN II revision of FORTRAN language  
1959 ATLAS computer designed - virtual memory invented  
1959 (approx) Move from valves to transistors; Batch Operating Systems (one program at a time)
- 1960 COBOL common business language - commercial programming language specification  
1960 (approx) portable handheld battery powered transistor radios become common  
1963 FORTRAN IV revision of FORTRAN scientific calculation programming language  
1964 (approx) Move from individual transistors to integrated circuits, Multiprogramming Operating System, more than 32k words of fast memory, faster than 1 microsecond (1MHz)  
1964 Gordon Moore observes IC chip densities doubling regularly  
1964 IBM System/360 - mainframe computer of choice - 18 000 units sold up to 1972  
1964 BASIC programming language system developed at Dartmouth College  
1965 revised COBOL 65 (became ANSI standard COBOL in 1968)  
1966 FORTRAN 66 (FORTRAN IV) ANSI standard - widely implemented version of scientific calculation programming language  
1968 NATO Software Conference identifies "the software crisis"  
1969 ARPANET network started (USA) - start of the Internet
- 1970 RAM chips appear - fast cheap mass production memory  
1970 (approx) pocket calculator drives out slide rule for personal scientific calculations  
1971 First microprocessor appears (4004); Altair 8080 microcomputer

- 1971 IBM System/370 - mainframe - total of 80 000 units sold up to 1988
- 1971 PROLOG - programming in logic language with strong mathematical foundations
- 1972 e-mail invented
- 1972 (approx) minicomputers - 12 or 16 bit computers - mainly scientific; cheaper than peripherals: DEC PDP-8
- 1972 Pascal programming language: portable initially for education
- 1972 C programming language - structured language suited for low-level portable systems programming of minicomputers
- 1973 First mass-market computer game (Pong)
- 1974 Personal computers appear: Altair hobbyists' personal micro-computer
- 1974 Microsoft founded - to sell BASIC standalone programming system for Altair
- 1974 COBOL 1974 revision of commercial programming language
- 1975 UNIX operating system marketed; C language starts to become popular
- 1975 (approx) XEROX invents Graphical User Interface (GUI)
- 1977 FORTRAN 77 revision with structured programming features
- 1977 software error in navigation controls of F-16 aeroplane: flips upside down when crosses equator (simulator)
- 1979 PET & TRS-80 user-friendly micro computers: 8 bit systems; CP/M operating system common
- 1979 VisiCalc - first computer spreadsheet program - for Apple II
  
- 1981 IBM PC released - 16 bit personal computer for desktops; command line operating system MS-DOS word processing becomes software application for PC rather than dedicated system
- 1981 (approx) DEC VAX-11 32 bit extended minicomputers as small mainframes
- 1981 the bug heard 'round the world - first Space Shuttle launch delayed 2 days by software bug
- 1983 Lotus 1-2-3 spreadsheet for 16 bit personal computers becomes popular application
- 1984 Apple Macintosh released: popular introduction of Graphical User Interface
- 1985 Windows GUI introduced for IBM PC
- 1985 FTP (file transfer protocol) specification published; widely implemented and used
- 1987 first widespread computer viruses and worms spread to thousands of computer systems
- 1988 programming design errors discovered in THERAC-25 therapeutic radiation machine: kills 3 people
- 1989 World Wide Web invented
- 1990 FORTRAN 90 revision of scientific programming language; with vector and parallel programming features
- 1991 (approx) Java programming language introduced - object oriented; network aware
- 1994 errors discovered after widespread release in Pentium processor chip
- 1995 WWW traffic overtakes FTP traffic on the Internet
- 1997 Deep Blue chess computer beats world champion Gary Kasparov at chess
- 1999 widespread fear of Y2K bug triggers millions of dollars of work to fix legacy systems
- 2000 First 1 GHz microprocessors
- 2000 worldwide annual PC sales reach 133 Million (all types)