



6.3 Types of program translator

Name: _____

Class: _____

Date: _____

Time: **234 minutes**

Marks: **185 marks**

Comments:

Q1.

Employees at a bank use client computers to access data that is stored on a database server.

The database server uses software to query and modify data stored in a database on hard disk drives. It returns the results of these queries to the clients over the bank's computer network.

The performance of the system is unsatisfactory: the time-delay between a client sending a query to the server and the client receiving the results is unacceptably long.

Explain how the performance of the system might be improved. You should consider the following factors that might be affecting the performance:

- the hardware of the server
- the design of the computer network
- the database and software running on the server.

In your answer you will be assessed on your ability to follow a line of reasoning to produce a coherent, relevant and structured response.

(Total 12 marks)

Q2.

- (a) First and second generation languages are known as low-level languages. What is meant by the term low-level language?

(1)

- (b) Programs written using a high-level language are easier to maintain and understand than programs written in a low-level language.

Describe **two** ways in which high-level languages can make this possible.

(2)

- (c) A student new to programming has heard that some languages are compiled and others are interpreted, and that compilers and interpreters are both known as types of translator.

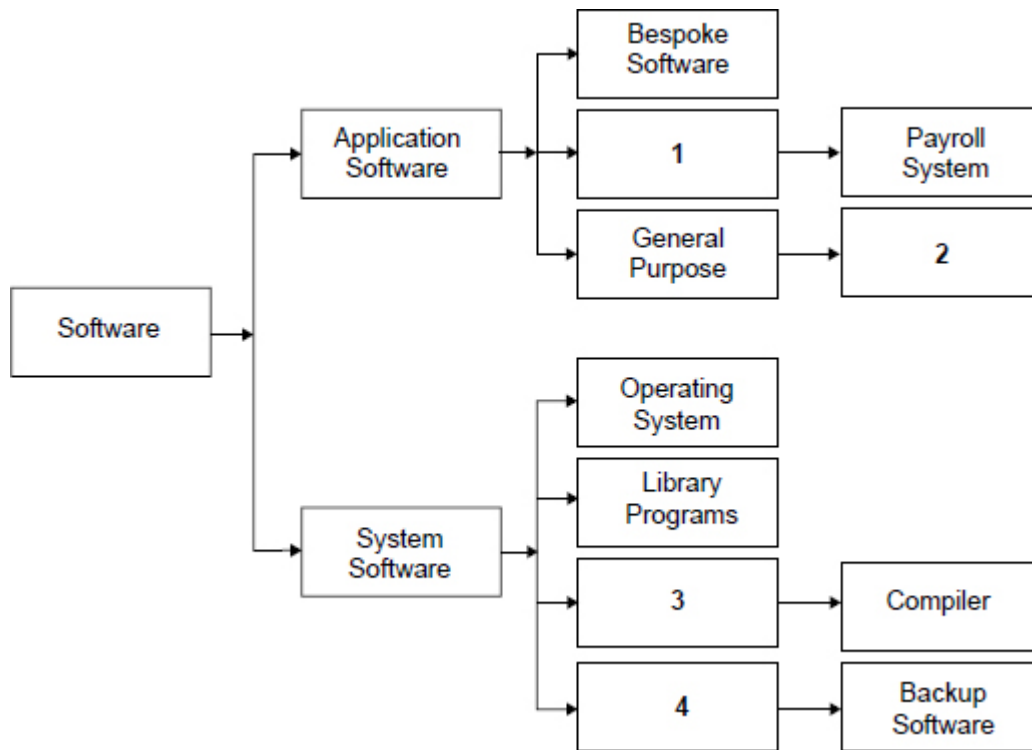
Describe to this student:

- In your answer, you will be assessed on your ability to use good English and to organise your answer clearly in complete sentences, using specialist vocabulary where appropriate.

同

(Total 9 marks)

The diagram below shows the classifications of various types of software used on a computer system and some examples of these types.



- (a) Complete the diagram by suggesting labels for the boxes numbered 1 to 4 in the diagram.

1. _____
2. _____
3. _____
4. _____

(4)

- (b) (i) Machine code is the first generation of programming language.

What is the second generation of programming language?

(1)

- (ii) A program written in a second generation programming language has been loaded into a computer. In this form it cannot be directly executed on this computer.

What has to be done to make an executable form of the program, which can be directly executed by this computer, and what would be used, typically, to do this?

(2)

- (iii) A programmer then finds that when the executable form of the program is transferred unaltered to another computer, the program does not run and an error message is displayed.

Why might the executable form of the program not be able to run on this computer?

(1)

(Total 8 marks)

Q4.

A school robotics club has recently purchased a robotics kit after the teacher in charge saw an advert in a magazine. The advert is reproduced below.

RoboEddy - a new educational robot

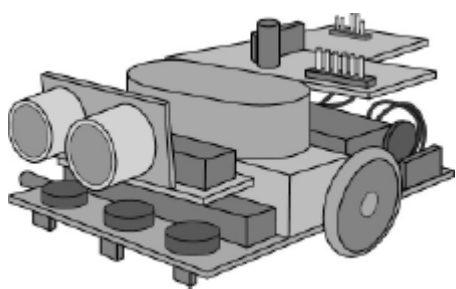
Specification

Hardware

- 500 Mhz processor
- 32 MB RAM
- 4 timers
- Wi-Fi communications via WLAN 802.11g radio
- Dual H-bridge motor driver

Software

- Built in interpreter for the high level language RobotC
- Directly run assembly code
- XMODEM protocol for reliable file transfer
- Support for various analogue and digital sensors



**Available
NOW!**

Only £199

- (a) Using the XMODEM protocol, students at the robotics club can copy a RobotC program prepared on a desktop computer to the robot.

What is meant by the term *protocol*?

(1)

- (b) The RobotC program that has been copied to the robot can be executed by the built-in interpreter.

How does a high level language interpreter work?

(2)

- (c) The robot processor is different in some ways from a processor in a desktop computer, but it still follows the stored program concept.

What is meant by the term *stored program concept*?

(3)

- (d) As well as using RobotC, it is also possible to program the robot using assembly language.

The motor driver uses memory locations to store the current speed of each motor. The left motor speed is stored in memory location 21 and the right motor speed is stored in memory location 22.

The following set of three assembly language instructions can be used to take basic control of the motors:

LOAD XX - load a value from memory location XX into the accumulator

ADD XX - add the value stored in memory location XX to the accumulator

STORE XX - store the value in the accumulator in memory location XX

Selecting from the set of three instructions above, write a sequence of instructions that will swap the current left motor speed with the current right motor speed. Your program may use memory location 23 for temporary storage.

(3)

- (e) The students develop a program that can sort coloured balls into piles but it is found that the program is not very effective.

With regards to touch and vision, state **three** factors why a robot may find a task, such as sorting coloured balls, a hard task whereas for a 4-year-old child it is a relatively easy one.

Factor 1 _____

Factor 2 _____

Factor 3 _____

(3)

- (f) The robot identifies the colour of the balls using a digital still camera component.

- (i) Describe the principles of operation of a digital still camera.

EXAM PAPERS PRACTICE

(3)

- (ii) The digital still camera component can take high resolution images but the students have chosen to program it to take low resolution images instead.

Give a reason why the students might have only used a low resolution.

(1)

(Total 16 marks)

Q5.

The diagram below shows program code developed using different generations of programming languages.

Program 1 (with comments)

```
//Calculate  
FirstVar := 47;  
SecondVar := FirstVar + 2;  
FourthVar := ThirdVar;
```

Program 2 (with comments)

```
AB2F ; Load value 2F into accumulator  
BC5D ; Store contents of accumulator at address 5D  
E402 ; Add value 2 to accumulator  
BCFF ; Store contents of accumulator at address FF  
AC61 ; Load accumulator with contents of address 61  
BC4A ; Store contents of accumulator at address 4A
```

- (a) What generation of programming language was used to write **Program 1**?

_____ (1)

- (b) Machine code can be represented in different numeric formats.

- (i) Which numeric format is used by the machine code program in **Program 2**?

_____ (1)

- (ii) State **one** reason for using this format.

_____ (1)

- (iii) The machine for which **Program 2** has been written has limited addressing capability.

What are the lowest and highest memory addresses that can be addressed by this machine?

Lowest address: _____

Highest address: _____ (1)

- (c) Give an example of a situation for which it would be appropriate to write a program in a low level language (ie machine code or assembly language).

_____ (1)

- (d) Explain the differences between a compiler and an interpreter

(4)
(Total 9 marks)

A programmer could use either an assembly language or a high level language to code programs for sale.

- # EXAM PAPERS PRACTICE

(2)

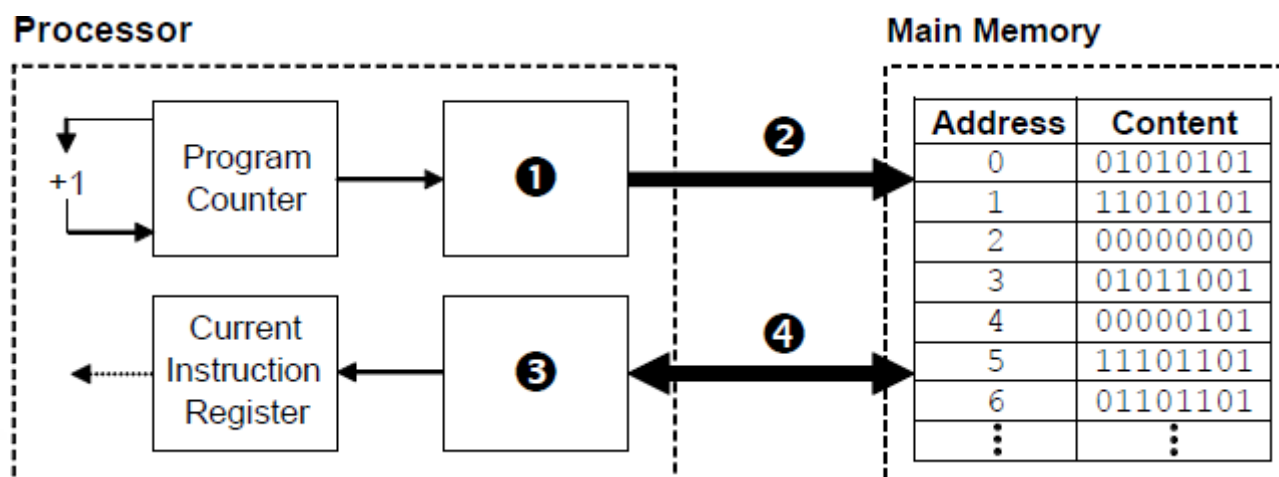
- Give **two** advantages of using a compiler, rather than an interpreter, to prepare a runnable program ready for sale.

Page 9 of 25

(2)
(Total 4 marks)

Q7.

The diagram below shows the processor registers and busses that are used during the fetch part of the fetch-execute cycle, together with the main memory. The values stored in memory locations 0 to 6 in the main memory are machine code instructions.



- (a) Name the components that are labelled with the numbers 1 to 4. In the case of register names, the full names must be stated.

Number	Component Name
1	
2	
3	
4	

(4)

- (b) Explain what happens during the decode and execute stages of the fetch-execute cycle.

(3)

- (c) The machine code instructions in the main memory in the diagram above are shown in binary.
When programmers look at machine code instructions they usually prefer to view them in hexadecimal.

State **one** reason why this is the case.

(1)

- (d) The machine code instructions in the main memory in the diagram above were produced when an assembly language program was translated into machine code.

- (i) What type of program translator was used to do this?

(1)

- (ii) Most computer programs are initially written in an imperative high level language rather than assembly language.

Explain why this is the case.

EXAM PAPERS PRACTICE

(3)

(Total 12 marks)

Q8.

Programs written in a high level language can be compiled or interpreted.

- (a) Companies that develop computer programs to sell always compile the final version of a program before distributing it to customers.

Explain why a compiler is used to produce the final version of a computer program.

(2)

- (b) Scripting programming languages can be used to write programs which are interpreted and executed in a web browser on any Internet user's computer.

Explain why programs written in a scripting language for this purpose are interpreted rather than compiled.

(2)

(Total 4 marks)

Q9.

Figures 1, 2 and 3 show three versions of the same program.

Figure 1	Figure 2	Figure 3
Total := 60 + 10;	LD #60	00101000 00111100
	ADD #10	00111000 00001010
	ST Total	01100000 00101001

- (a) What type of translator program would be used to convert **Figure 1's** program into **Figure 3's** program?

(1)

- (b) What type of translator program would be used to convert **Figure 2's** program into **Figure 3's** program?

(1)

(Total 2 marks)

Q10.

- (a) The high-level language statement

A := B + 5

is to be written in assembly language.

Complete the following assembly language statements, which are to be the equivalent of the above high level language statement. The Load and Store

instructions imply the use of the accumulator register.

Load _____

_____ #5

Store _____

(3)

- (b) (i) What type of translator is required to translate assembly code statements into machine code?

(1)

- (ii) What type of translator is required to translate a high-level language statement into machine code?

(1)

(Total 5 marks)

Q11.

A programmer developing a new application will choose a high level programming language. Typical modern software will provide an 'environment' in which all aspects of the application's development can be done.

- (a) What feature must all program development environments have for the initial creation of the program code?

(1)

- (b) The next stage in the development may use a **compiler**.

Explain what a compiler does, and suggest **one** output which may result from the compilation process, other than the executable code.

Output: _____

(3)

- (c) An alternative to a compiler is an **interpreter**.

Explain how an interpreter attempts to run a program.

(2)

(d) The program development environment has **both** a compiler and interpreter.

(i) Give **one** advantage of using an interpreter.

(1)

(ii) Give **one** advantage of using a compiler.

(1)

(Total 8 marks)

Q12.

Figure 1 and **Figure 2** show two different versions of a **small section of a program**.

Figure 1

Load	113,	R1
Load	114,	R2
Load	115,	R3
Add	R1,	R2
Add	R3,	R2
Store	R2,	160
End		

Figure 2

(a)	Main memory	
100	10000000	01110001
101	10000001	01110010
102	10000010	01110011
103	11110000	00000000
104	11110011	00000000
105	00010001	10100000
106	00000000	00000000

(a) In **Figure 2** the label is missing from the column showing 100 to 106.

What should this label be? _____

(1)

(b) What generation of programming language is shown in **Figure 1**?

(1)

(c) The code as written by the programmer is shown in **Figure 1**. A translator program is needed to produce a version of the code the processor can execute.

(i) What is this translator program called?

_____ (1)

- (ii) Following the translation process various outputs will be produced.
One output from this translation is the machine code executable file.

Name **one** other possible output _____ (1)

- (d) Many digital computers operate on the stored program concept.

Explain the *stored program concept*.

(4)
(Total 8 marks)

Q13.

The figure below shows three different programs which have been developed using different generations of programming language.

Program 1	Program 2	Program 3
<pre>If Sales > 10000 Then BonusPayment :=True etc. etc. Procedure InputNewData Procedure ToOutputFile etc ...</pre>	<pre>Move #0, R1 Add R1, R2 Store R1, 0197 Move 0198, R3 Add R2, R3 Cmp R3, #1662 Bne 0988 etc ...</pre>	<pre>1000 0101 1010 1111 1010 1111 1110 0001 1010 1111 etc ...</pre>

The above programs were written for different tasks.

- (a) What generation of programming language was used for Program 1?

_____ (1)

- (b) Indicate which program was most likely to have been written for:

- (i) controlling a new hardware device.

_____ (1)

- (ii) a payroll application.

(1)

- (c) Program 1, Program 2 and Program 3 may require translation before each can be executed.

	Assembler	Compiler	None
Program 1			
Program 2			
Program 3			

Put **one** tick on each row in the table above to indicate the translator software required.

(3)

- (d) Describe how **interpreter** software enables a program written in a high level language to be executed.

(2)

- (e) A friend gives you a copy of a freeware **assembler**. Why might you not be able to use this successfully on your computer?

EXAM PAPERS PRACTICE

(1)

(Total 9 marks)

Q14.

- (a) (i) Explain **one** difference between a procedure and a function.

(2)

- (ii) Name and describe a built-in function you have used in your programming work, or when using a generic software package.

(2)

- (b) A particular built-in function is described in a programming language's help files as follows:

Function MatchString(ThisString , StringSearchedFor : String) : Boolean

The function **MatchString** returns a **Boolean** value indicating whether or not the string **StringSearchedFor** appears within the string **ThisString**.

An error is returned when a function call is incorrectly formed.

What value is returned to the Result1, Result2 and Result3 variables from the following function calls?

- (i) Result1 := MatchString ('Harry Potter', 'Pot')

_____ (1)

- (ii) Result2 := MatchString ('Potter', 'Harry Potter')

_____ (1)

- (iii) Result3 := MatchString ('Harry Potter', 59)

_____ (1)

- (c) In part (b) (i) Result1 is an identifier used for a variable. Name **two** other uses for identifiers in a high level language.

1. _____ (2)

2. _____

- (d) The programming language being used has both compiler and interpreter software for program development.

Give **one** advantage of the use of each.

Interpreter advantage _____

_____ (1)

Compiler advantage _____

_____ (1)

(1)
(Total 11 marks)

Q15.

Figure 1 and **Figure 2** below show **two** versions of the same program.

Figure 1			Figure 2	
			(c)	
Move	#45,	R0	100	00101000 00101101
Move	#4,	R1	101	00101001 00000100
Move	#96,	R2	102	00101010 01100000
Add	R2,	R1	103	10100001 00000000
Add	R1,	R0	104	10100000 00000000

- (a) What generation of programming language is shown in **Figure 1**?

_____ (1)

- (b) What generation of programming language is shown in **Figure 2**?

_____ (1)

- (c) What would be a suitable heading for the column labelled (c) in **Figure 2**?

_____ (1)

- (d) What software will be needed to translate the program code shown in **Figure 1** to the program code shown in **Figure 2**?

EXAM PAPERS PRACTICE _____ (1)

- (e) What is the relationship between the program instructions shown in **Figure 1** program instructions in **Figure 2**?

_____ (1)

- (f) In addition to the executable file, what output could the software referred to in part (d) produce?

_____ (1)

(Total 6 marks)

Q16.

(a) Give **one** example of a

(i) first generation programming language

_____ (1)

(ii) second generation programming language

_____ (1)

(iii) third generation programming language

_____ (1)

(b) Give **two** advantages of programming in third generation programming languages, rather than in the previous two generations.

1. _____
2. _____ (2)

(c) Third generation programming languages may be compiled or interpreted. Describe the process performed by

(i) a compiler

_____ (2)

(ii) an interpreter

_____ (2)

(d) When would it be appropriate to use **each** of the following?
In **each** case give the reason for your choice.

(i) a compiler

Use _____
Reason _____ (2)

(ii) an interpreter

Use _____

Reason _____

(2)

(Total 13 marks)

Q17.

There are a large number of programming languages. System software such as *compilers*, *assemblers*, *interpreters* are used to translate programs into machine instructions.

(a) Explain the different ways in which a compiler and an interpreter operate.

(i) a compiler _____

(1)

(ii) an interpreter? _____

(1)

(b) If both a compiler and an interpreter are available for a particular programming language, under what circumstances would it be preferable to use:

(i) a compiler; _____

(1)

(ii) an interpreter; _____

(1)

(c) In what way does an assembler differ from a compiler?

(1)

(Total 5 marks)

Q18.

(a) Machine code is the first generation of programming languages. All other generations of programming languages need a program translator before the program can be executed. Name a type of translator suitable for:

(i) Second generation language programs:

(1)

(ii) Third generation language programs:

(1)

- (b) Imperative *high level languages* are third generation.

Give **two** characteristics of high level languages that distinguish them from second generation languages.

1. _____

2. _____

(2)

- (c) In one high level language an example of a constant definition would be

CONST VatRate = 17.5;

State **one** advantage of using a named constant, like VatRate, rather than the actual value (17.5) in a high level language program.

(1)

- (d) (i) Name an imperative high level language which you have studied.

(1)

For the language you have named in (d) (i) above, give an example, using the correct syntax, of:

(ii) iteration: _____

EXAM PAPERS PRACTICE

(2)

(iii) selection: _____

(2)

(Total 10 marks)

Q19.

- (a) Some of the basic components of a computer system are processor, main memory, and secondary storage.

- (i) What connects the processor and main memory?

(1)

(ii) What is the purpose of secondary storage?

(1)

(iii) Describe what happens during the fetch-execute cycle.

(2)

(b) (i) Machine code is the first generation programming language. What is the second generation?

(1)

(ii) A programmer writes a program in a second generation programming language.
What has to be done to this program before it can be executed?

(2)

EXAM PAPERS PRACTICE

(iii) Some high level languages are classified as *imperative*. What is meant by imperative?

(1)

(iv) Give an example of an imperative high level language.

(1)

(v) What is the relationship between an imperative high level language statement and its machine code equivalent?

(1)

- (vi) Give **two** disadvantages of programming in first and second generation programming languages compared with imperative high level languages.

1. _____

2. _____

(2)

(Total 12 marks)

Q20.

- (a) Programmers are encouraged to adopt a structured approach to writing programs. One reason is so that programmers can write code which can be more easily understood by another programmer.

Explain **two** other reasons.

1. _____

(2)

2. _____

(2)

- (b) Give **three** features of an imperative high level programming language which allow programmers to write “easy-to-understand” code.

1. _____

EXAM PAPERS PRACTICE

2. _____

3. _____

(3)

- (c) Distinguish between a compiler and an interpreter.

(2)

Q21.

- (a) Contrast low-level and high-level programming languages.

(4)

- (b) Describe **one** situation where a high-level language is inappropriate but a low-level language could be used.

(2)

(Total 6 marks)

EXAM PAPERS PRACTICE

Q22.

A particular computer language is supplied with an interpreter and a compiler.

- (a) Give **one** reason for using the interpreter during development.

(1)

- (b) Give **two** reasons for using the compiler to translate the completed program.

(2)

Q23.

Assemblers and *compilers* are two kinds of program translators. Write **two** statements about each of them to distinguish one from the other.

An assembler

1. _____

2. _____

A compiler

1. _____

2. _____

(Total 4 marks)

