EXAM PAPERS PRACTICE

6.2 Classification of programming languages

Name:	
Class:	
Date:	

Time:	**197 minutes**
Marks:	**164 marks**
Comments:	

Q1.

Cameras within a taxi take still images once every second for security purposes. The images are compressed using run-length encoding and stored on a flash memory card within the camera.

Describe how a digital image could be captured by a digital camera and compressed using run-length encoding.

____

____

____

____

____

____

____

____

____

____

____

____

(Total 6 marks)

Q2.

Figure 1 shows the structure of an example machine code instruction, taken from the instruction set of a particular processor.

Figure 1

Opcode							Operand(s)							
Basic Machine Operation						Addressing Mode								
0	1	1	0	1	0	1	0	0	1	0	1	0	1	1

(a) How many different basic machine operations could be supported by the instruction set of the processor used in the example in **Figure 1**?

____

____

(1)

Figure 2 shows an assembly language program together with the contents of a section of the main memory of the computer that the program will be executed on.

The assembly language instruction set that has been used to write the program is listed in **Table 1**.The lines of the assembly language program have been numbered to help you answer question parts (**b**) to (**d**)

Figure 2

Line	**Command**
1	`LDR  R2,   #100`
2	`LDR  R3,   101`
3	`ADD  R2,   R2,   R3`
4	`LSL  R3,   R2,   #1`
5	`HALT`

Memory Address (in decimal)	**Main Memory Contents (in decimal)**
100	23
101	10
102	62
103	18

(b) What value will be stored in register `R2` immediately after the command in line 1 has been executed?

____

(1)

(c) What value will be stored in register `R2` immediately after the program has executed the commands from line 1 through to line 3?

____

(1)

(d) What value will be stored in register `R3` after the complete program has finished executing?

____

Table 1

`LDR Rd, <memory ref>`	Load the value stored in the memory location specified by `<memory ref>` into register `d`.
`STR Rd, <memory ref>`	Store the value that is in register `d` into the memory location specified by `<memory ref>`.
`ADD Rd, Rn, <operand2>`	Add the value specified in `<operand2>` to the value in register `n` and store the result in register `d`.
`SUB Rd, Rn, <operand2>`	Subtract the value specified by `<operand2>` from the value in

	register n and store the result in register d.
`MOV Rd, <operand2>`	Copy the value specified by `<operand2>` into register d.
`CMP Rn, <operand2>`	Compare the value stored in register n with the value specified by `<operand2>`.
`B <label>`	Always branch to the instruction at position `<label>` in the program.
`B<condition> <label>`	Conditionally branch to the instruction at position `<label>` in the program if the last comparison met the criteria specified by the `<condition>`. Possible values for `<condition>` and their meaning are: • EQ: Equal to. • NE: Not equal to. • GT: Greater than. • LT: Less than.
`AND Rd, Rn, <operand2>`	Perform a bitwise logical AND operation between the value in register n and the value specified by `<operand2>` and store the result in register d.
`ORR Rd, Rn, <operand2>`	Perform a bitwise logical OR operation between the value in register n and the value specified by `<operand2>` and store the result in register d.
`EOR Rd, Rn, <operand2>`	Perform a bitwise logical exclusive or (XOR) operation between the value in register n and the value specified by `<operand2>` and store the result in register d.
`MVN Rd, <operand2>`	Perform a bitwise logical NOT operation on the value specified by `<operand2>` and store the result in register d.
`LSL Rd, Rn, <operand2>`	Logically shift left the value stored in register n by the number of bits specified by `<operand2>` and store the result in register d.
`LSR Rd, Rn, <operand2>`	Logically shift right the value stored in register n by the number of bits specified by `<operand2>` and store the result in register d.
`HALT`	Stops the execution of the program.

Interpretation of `<operand2>`

`<operand2>` can be interpreted in two different ways, depending upon whether the first symbol is a # or an R:

- # - use the decimal value specified after the #, eg #25 means use the decimal value 25.
- Rm - use the value stored in register m, eg R6 means use the value stored in register 6.

The available general purpose registers that the programmer can use are numbered 0 to 12.

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

Companies that develop computer programs to sell usually compile the final version of a program before distributing it to customers.

(e) Explain why the final version of a computer program is usually translated using a compiler.

____

____

____

____

(2)

(f) The JavaScript programming language can be used to write programs that are executed in a web browser on any Internet user's computer.

Explain why programs written in the JavaScript language, to be executed in a web browser, are interpreted rather than compiled.

____

____

____

____

(2)

(Total 8 marks)

Q3.

(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:

- the role of a translator
- the differences between a compiler and an interpreter
- a situation where you would use a compiler
- a situation where you would use an interpreter.

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.

Use the space provided to write your answer to this question

____

____

____

____

____

____

____

____

____

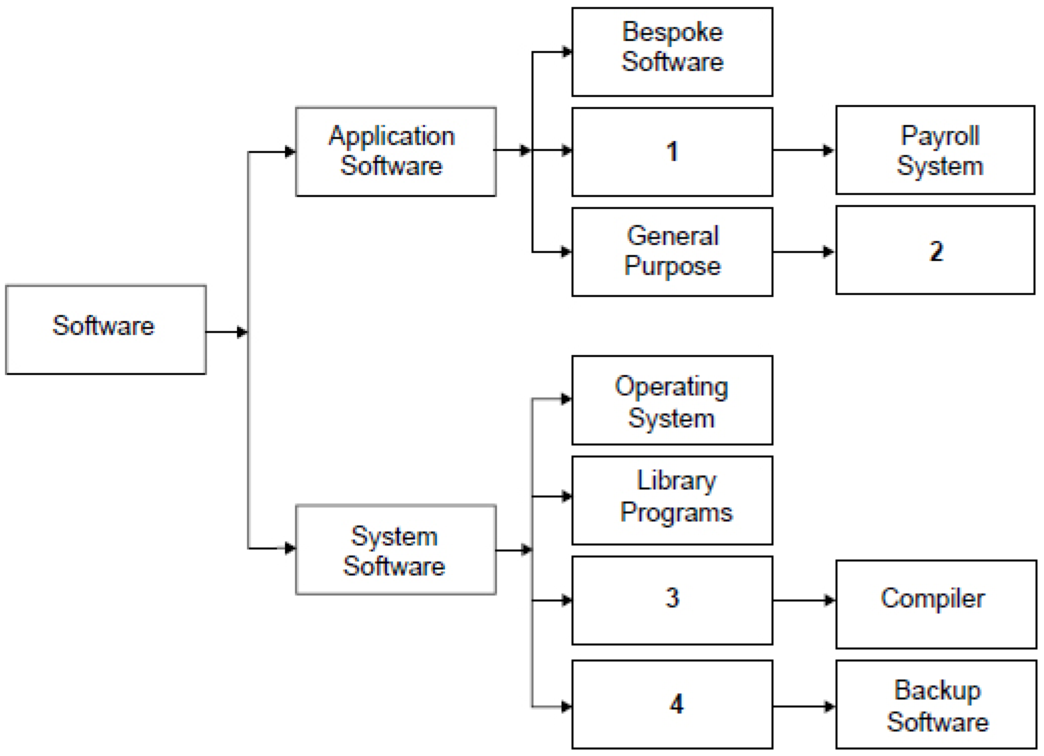
____

(6)

(Total 9 marks)

Q4.

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)

Q5.

For each problem in the table below, place a tick in the appropriate box to indicate the generation of programming language best suited to developing a solution to the problem.

Do **not** tick more than one box in each row.

	Generation		
Problem	**1st**	**2nd**	**4th**
Developing a diagnosis program for medical symptoms			
Developing a program for an embedded microprocessor for a washing machine			

(Total 2 marks)

Q6.

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

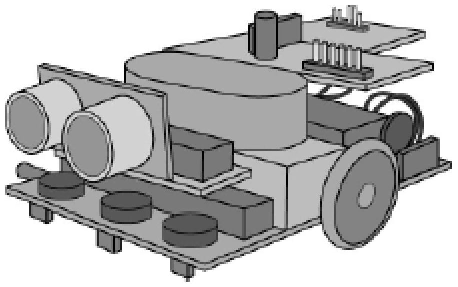
Specification

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.

____

____

____

____

____

____

(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)

Q7.

When writing a program, a programmer could use an assembly language, a high level imperative language or a high level declarative language.

Outline the major differences in each of these **three** approaches. For each language type, your answer could include:

- advantages and disadvantages compared to other language types
- how the programmer would express their programs
- what translation software could be used, if applicable
- a situation where it might be the most appropriate choice.

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.

(Total 8 marks)

Q8.

There are many third generation programming languages available. Some of these can be classified as imperative high level languages.

(a) Explain what is meant by the term *imperative high level language*?

____

____

____

____

(2)

(b) Give **one** reason for there being so many third generation programming languages.

____

____

____

____

(1)

(Total 3 marks)

Q9.

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)

Q10.

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

Figure 1			**Figure 2**		
(x)	**(y)**	**(z)**	**(x)**	**(y)**	**(z)**
`200`	`LOAD`	`7`	`200`	`01010110`	`00000111`
`201`	`ADD`	`3`	`201`	`11010000`	`00000011`
`202`	`ADD`	`6`	`202`	`11010000`	`00000110`
`203`	`STORE`	`255`	`203`	`11110000`	`11111111`

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

____

(1)

(b) In both figures above there is a column labelled **(x)**.

What would be a suitable heading for this column?

____

(1)

(c) In both tables the instruction is split into two parts.

What are the names of the instruction parts in columns **(y)** and **(z)**?

(y) ____

(z) ____

(2)

(d) What is the relationship between the instructions in **Figure 1** and **Figure 2**?

____

____

(1)

(Total 5 marks)

Q11.

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

(a) Give **two** limitations of using assembly language to code a program.

1. ____

____

2. ____

____

(2)

(b) If a program is coded using a high level language, then either a compiler or an interpreter will need to be used.

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

1. ____

____

____

2. ____

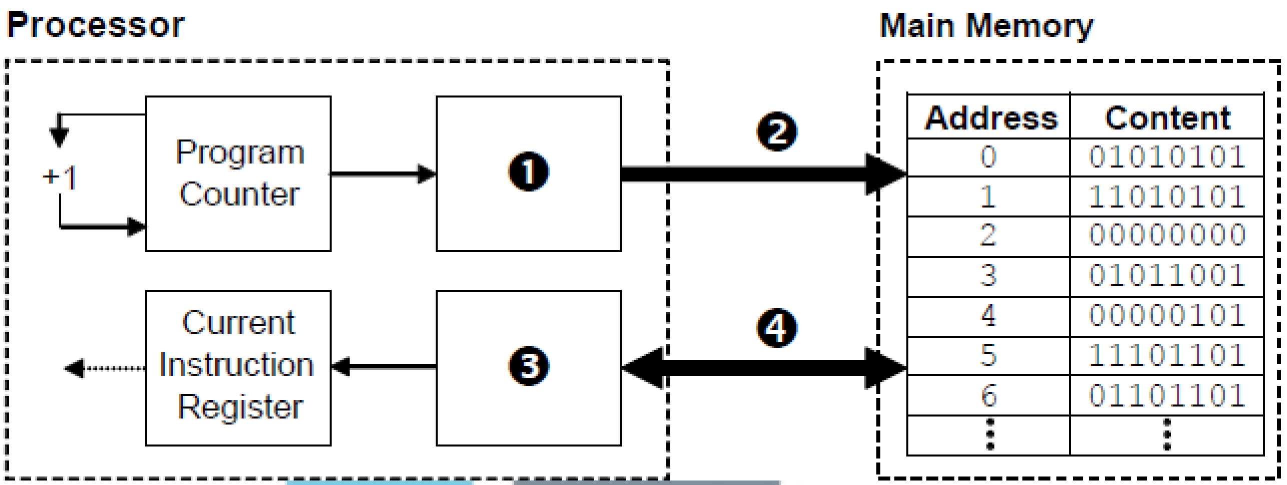
____

____

(2)
(Total 4 marks)

Q12.

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.



Address	Content
0	01010101
1	11010101
2	00000000
3	01011001
4	00000101
5	11101101
6	01101101
⋮	⋮

(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
❶	
❷	
❸	
❹	

(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.

____

____

____

____

____

____

(3)

(Total 12 marks)

Q13.

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)

Q14.

High level programming languages can be classified as being either imperative or declarative.

(a) Explain what is meant by a *declarative language.*

____

____

____

____

(1)

(b) Name **one** type of application for which a declarative language could be used.

____

____

(1)
(Total 2 marks)

Q15.

A computer system has the following assembly code instructions that you are to use in this question:

Label	**Opcode**	**Operand (s)**	**Description**
	AND	#nn	Logical AND the accumulator with hexadecimal value nn
	OR	#nn	Logical OR the accumulator with hexadecimal value nn
	LD	nnnn	Load contents of hexadecimal address nnnn into the accumulator

	LD	label	Load contents of labelled memory into the accumulator
	ST	nnnn	Store contents of the accumulator into hexadecimal address nnnn
	ST	label	Store contents of the accumulator into labelled memory
	ADD	#nn	Add the hexadecimal value nn to the accumulator
	ADD	nnnn	Add the contents of hexadecimal address nnnn to the accumulator
	MUL	#nn	Multiply the accumulator by the hexadecimal value nn
	MUL	nnnn	Multiply the accumulator by the contents of the hexadecimal address nnnn
	CMP	#nn	Compare the accumulator with hexadecimal value nn
	CMP	label	Compare the accumulator with the contents of the labelled memory
	JP	label	Jump unconditionally to the label
	JE	label	Jump to the label if the result of a compare shows the accumulator to be equal to the operand
	JNE	label	Jump to the label if the result of a compare shows the accumulator not to be equal to the operand
	JG	label	Jump to the label if the result of a compare shows the accumulator to be greater than the operand
	JGE	label	Jump to the label if the result of a compare shows the accumulator to be greater than or equal to the operand
	JL	label	Jump to the label if the result of a compare shows the accumulator to be less than the operand
	JLE	label	Jump to the label if the result of a compare shows the accumulator to be less than or equal to the operand

(a) Give **two** reasons why some software is still developed in an assembly language.

1. ____

____

2. ____

____

(2)

(b) Give **one** reason why the majority of software is no longer developed using assembly language.

(1)
(Total 3 marks)

Q16.

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

Program 1

```
If Sales > 10000
  Then BonusPayment :=True
etc.
etc.

Procedure InputNewData
Procedure ToOutputFile

etc ...
```

Program 2

```
Move    #0,  R1
Add     R1,  R2
Store   R1,  0197
Move    0198, R3
Add     R2,  R3
Cmp     R3,  #1662
Bne     0988

etc ...
```

Program 3

```
1000 0101
1010 1111
1010 1111
1110 0001
1010 1111

etc ...
```

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?

___

___

(1)

(Total 9 marks)

Q17.

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

Figure 1

Move	#45,	R0
Move	#4,	R1
Move	#96,	R2
Add	R2,	R1
Add	R1,	R0

Figure 2

(c)	
100	00101000 00101101
101	00101001 00000100
102	00101010 01100000
103	10100001 00000000
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**?

___

(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)

Q18.

(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)

Q19.

(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: ____

____

____

(2)

(iii) selection: ____

____

____

(2)

(Total 10 marks)

Q20.

(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)

(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)

Q21.

Programming languages are subdivided into generations. An imperative high level language is a third generation language.

(a) Name the language type for:

(i) first generation; ____

(1)

(ii) second generation. ____

(1)

(b) Name **one** specific example of a third generation programming language.

____

(1)

(Total 3 marks)

Q22.

(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)

Q23.

(a) Explain what is meant by a *programming language*.

____

____

____

____

(2)

(b) Distinguish between *low-level* and *high-level* programming languages.

____

____

____

____

(2)

(c) What must happen to the source code of a program before it can be executed, and why is this necessary?

____

____

____

____

(2)

(Total 6 marks)