# System Software

Papers Dock

COMPUTER SCIENCE 9618 PAPER 1

# System Software


**A software platform that provides facilities for programs to be run which are benefit to a user.**

## What is the purpose of the operating system?

- To hide the complexities of the hardware from the user
- To provide a platform for software to run
- To provide a user interface


- User types in instructions to open or launch a program.
- User is in direct communication with the computer system.
- Usually a number of instructions are needed to be typed in to open or launch a program.

- User interacts with the computer system by using icons
- User does not need to know where application resides within computer
- User launches application by the use of pointer
- Windows is the example of G.U.I

# Explain the key management tasks carried out by the operating system?

## 1) Memory Management

- controls the movement of data between RAM, processor and Virtual Memory
- allocates memory to processes
- ensure fair usage of memory
- organize memory by making use of Virtual Memory
- Keeps process separate
- To release memory when a process stops




## 2) Process Management

- Decides which process to run next and manages their scheduling
- support multitasking
- Handles priorities
- Enable process to share information
- prevents interference between the processes
- Manages which resources the process requires

## 3) Security Management

- Create accounts/passwords and sets up user account
- Provides firewall or anti malware
- Automatic Backup
- Access Rights

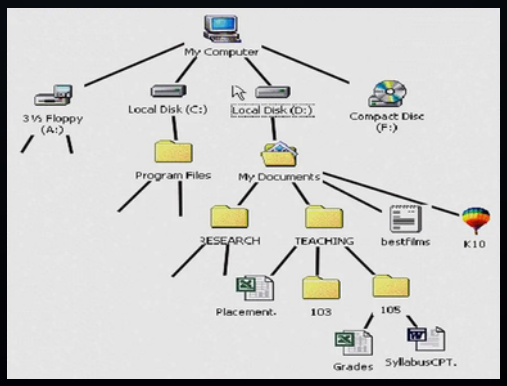
## 4) File Management

- Create file / folders
- Rename files / folders
- storage space divide into file allocation unit
- space allocated for particular files
- Maintains directory structure
- Specific tasks that can be performed on a file e.g open, delete, copy
- Specifies the logical method of file storage
- Provides file naming convention
- controls file access

Allocation Unit

Allocation unit size represents the smallest amount of disk space that can be used to hold a file

Directory Structure



## 5) Provision of user interface

- Allows a user to communicate with hardware by making navigation around the system easier.
- Provides facilities for user to input data.
- Provides facilities to show output results to user.
- E.g. CLI and GUI

## 6) Printer Management

- Installs printer driver
- Sends data to the printer
- Handles error messages
- Sends commands to the printer

## 7) Interrupt Handling

- Identifies priorities to the interrupt.
- Saves data on power outage
- Loads appropriate interrupt service routine.

## 8) Input Output Management

- Installation of appropriate drivers
- Control access to data being sent
- Control access to hardware
- Manages communication between devices

## 9) Hardware Management

- Receives data from input devices / sends data to output devices
- use of device driver

## Describe the ways in which memory management organizes and allocates Random Access Memory (RAM)

- RAM is assigned into blocks
- Dynamic allocation of RAM to programs / processes
- Reclaims unused blocks of RAM
- Prevents two programs / processes from occupying the same area of RAM at the same time
- manages paging, segmentation, and virtual memory

## Describe how the operating system manages the peripheral hardware devices of the computer ?

- Installs device drivers
- To allow communication between peripherals and computer
- Sends data and receives data to and from peripherals
- Such as to an output device and from an input device
- Handles buffers for transfer of data
- To ensure smooth transfer between devices that transmit and receive at different speeds
- Manages interrupts / signals from the device

## Describe how the operating system manages processes ?

- Manages the scheduling of processes
- Manages which resources the processes require
- Such as allocating memory
- Enables processes to share data
- Prevents interference between processes
- Handles the process queue
- It allows multi-tasking
- By ensuring fair access, handling priorities and handling interrupts

## Explain how memory management and process management support multi-tasking ?

Memory management

- Stores data from all currently running programs concurrently in RAM
- Stops the data from overwriting each other in RAM/primary storage
- Decides which processes should be in main memory
- Makes efficient use of memory
-

Process management

- Allows one process to be paused whilst another process can be actioned
- Decides which process is to be run next
- Switches between processes to allow them to share the use of the processor
- Identification/description of scheduling

# Utility Software

**Utility softwares analyze and maintain a computer system and make it functional**

## Describe the purpose of utility software in a computer ?

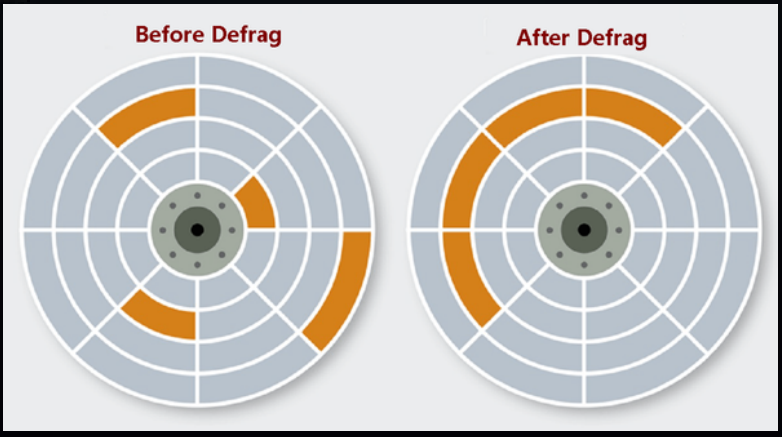
- **To help users to set-up, configure, analyze, optimize or maintain the computer**
- **by making memory allocation more efficient**
- **or by checking the system for faults**

## 1) Harddisk Defragmentation

- **Re-organizes the disk**
- **moves split file so they are contiguous**
- **creates a large area of contiguous free space**

### How this improves the performance ?

less time is taken to access the files because each one is contiguous so there is less head movement



## 2) Disc Content Analysis / Disk Repair

- Check for any error
- resolves any errors on the disk
- retrieves file from damaged disk
- marks bad sector of the disk

### How this improves the performance ?

preventing bad sectors being used because it identifies and mark them reduces access time by optimizing storage

## 3) Virus Checker

- scans files on a computer for malicious code
- scans files when they enter the system like when memory stick is inserted.
- sets up a schedule for virus checking
- delete virus
- regularly updates virus definition.



### How this improves the performance ?

makes more RAM available for programs to run because it removes software that might be taking up memory / replicating

## 4) File Compression Utility

- compress and decompress files
- infrequently used files are compressed
- saves space in harddisk.

### How this improves the performance ?

File compression utility improves performance by reducing the size of infrequently used files, which saves disk space and speeds up data retrieval. This allows more efficient storage and faster access to active files.

## 5) Backup Software

- creates a copy of the content of the disk.
- can be set up to automatically backup
- allows user to decide what is backed up
- allows off site backup
- may encrypt backed up files
- restores the data if necessary.

### How this improves the performance ?

A backup utility improves performance by safeguarding data, enabling quick recovery in case of data loss, and minimizing downtime. It also ensures data integrity, allowing systems to run smoothly without data-related disruptions.

## 6) Harddisk Formatter

- makes existing data inaccessible
- partition the disk into logical drives (C: Drive, D: Drive)
- prepare the disk for initial use
- might search for error
- sets up specified file system.

### How this improves the performance ?

A hard disk formatter improves performance by clearing old data, reducing fragmentation, and setting up an optimized file system, which leads to faster data access and efficient storage management.

## Explain how defragmentation can improve the performance of the computer ?

- **Rearranges blocks of individual files (on the HDD) so they are contiguous**
- **Accessing each file is faster**
- **Because there is no need to search for the next fragment / block of the file**
- **So less head movement is needed**

## Describe the reasons why a hard disk formatter is needed for the new hard disk?

- **Disk needs to be prepared for initial use**
- **Disk needs to be checked for errors**
- **A new file system needs to be generated on the disk**
- **The file allocation table needs to be set up**

# Program Library

## What is meant by a library routine / Program Library ?

- Pre-existing or Pre-compiled code
- Can be called in other programs
- by importing the library
- To perform common complex tasks

## Benefits of Library routine

- Less code needs to be written so saves time
- Pretested so reduces testing time and more likely to work
- Can be written in different programming languages that enables you to use special features of that language
- Can be complex algorithms and no need to work out how to write them
- Simplifies the program since just the name of the function is included
- If there is a an improvement in the library routine the program updates automatically

## Drawbacks of Library

- Compatibility issues; may not work with other code
- Not guaranteed through testing; may contain unknown unexpected bugs
- The code may not meet exact needs; may give unexpected results

**Describe how a program library can be used while writing a computer program ?**

- Program libraries store pre-written functions and routines
- The program library can be referenced/imported
- The functions/routines can be called in her own program

# Dynamic Link Library



- A collection of self- contained shared library program
- That are already compiled
- Linked to main program during execution
- Library program code is separated from .Exe file
- Library file only loaded into memory when required at run-time
- A DLL file can be made available to several application at the same time
- If a change is made in DLL code then all of the program that uses DLL will get changed .

## Benefits Of Dynamic Link Library

- The executable file is smaller because it does not contain all the library routines.
- DLL file is only loaded into memory when required at runtime, so main memory requirements for programs are reduced.
- A single DLL file can be made available to several applications, saving space in memory.
- Maintenance is not needed to be done by the programmer because the DLL is separate from the program.
- No need to recompile the main program when changes are made to DLL because changes / improvements / error correction to the DLL file code are done independently of the main program.

## Drawbacks Of Dynamic Link Library

- The executable code is not self-contained
- The DLL file needed to be included at run time
- Appropriate linking software must be available at run time to import the DLL file
- The DLL file should be present or else error
- Unexpected changes to DLL could mean the program stop working as expected
- Malicious changes to DLL file could install a virus on user's computer.

# Language Translator

Assembler Compiler Interpreter

## Operation Of Compiler

- Compiler translates high level language into machine code for the processor to execute
- Attempts to translate the whole source code
- Creates a separate error report at the end of the translation process
- If translation successful / no errors creates an executable file
- No need to give access to source code so makes it more difficult for the user to modify the code

## Operation Of Interpreter

- Interpreter translates high level language into machine code for the processor to execute line by line
- Reads each line then translates it and executes it
- Stops when an error is encountered // displays errors where it finds them
- The interpreter analyses and checks each line before executing

## Benefits Of Compiler

- Produces an executable file
- User does not have access to source code
- It will probably be faster to run the executable file
- Code does not have to be compiled each time it's executed
- Does not need the compiler to be present at run-time

## Drawbacks Of Compiler

- larger amounts of source code take time to compile
- slower to produce the object code than an interpreter
- code cannot be changed without recompilation
- the program will not run if there are any errors and the source code must be 100% correct for executable file to be produced
- errors cannot be corrected in real-time
- one error may result in other false errors being reported
- cannot easily test specific sections of the source code // cannot easily test unfinished source code

## Benefits Of Interpreter

- Errors can be corrected as they occur
- can run a partially complete program when developing
- The effect of any change made to the code can be seen immediately

# Drawbacks Of Interpreter

- No executable file is produced
- user has access to source code
- Need to translate the source code again and again so consumes time
- Interpreter should be present at the time of execution

## Differences between Compiler and Interpreter?

- Compiler creates an executable file
- Interpreter does not create executable file

- The compiled program can be independently distributed

- Interpreter executes each statement immediately after decoding
- Compiler checks the whole program for errors

- The interpreter software must be present in main memory every time the program is executed
- The compiled program does not require compiler

- Cross compilation is possible (compile on one hardware to run on another)

## Explain the purpose of language translator?

- To convert a higher level programming language to a different form.

**Describe the ways in which Jennifer can use both a compiler and an interpreter while developing the program.?**

**Interpreter:**

- **Use an interpreter while writing the program**
- **to test/debug the partially completed program**
- **because errors can be corrected and processing continue from where the execution stopped // errors can be corrected in real time // errors are identified one at a time**

**Compiler:**

- **Use the compiler after the program is complete**
- **to create an executable file**
- **Use the compiler to repeatedly test the same (completed) section**
- **without having to re-interpret every time // compiler not needed at run-time**



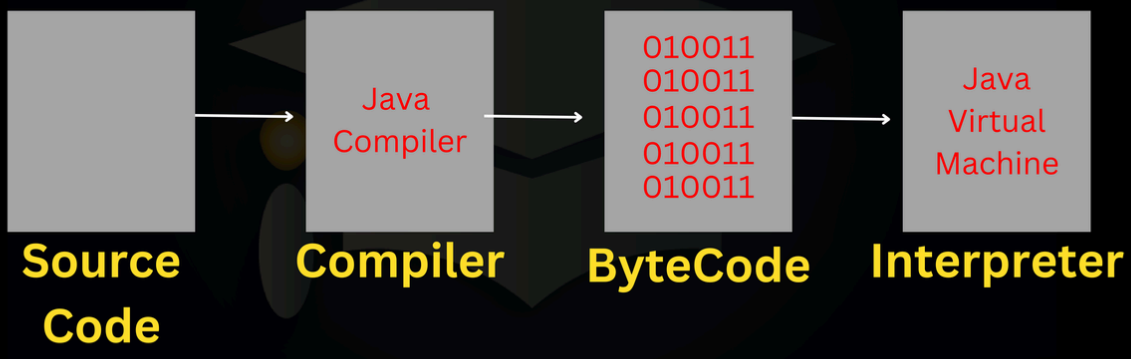
**Explain the reasons why the need to create an executable file makes the compiler the appropriate choice when the program is complete.?**

- **IProgram can be distributed without source code**
- **So it cannot be edited/stolen/plagiarised**
- **Users do not require the translator to run the program**
- **So time is not spent retranslating by the user**

# Partially Compiled And Partially Interpreted

**To combine the advantages of Compilation and Interpretation. We Partially compile and partially interpret this is known as Two-Way translation process**

## Java Source Code Translation




**Describe how a java source code program is translated?**

- Java uses a tow-step translation process
- Java code is partially compiled and partially interpreted
- Code is translated first into bytecode using the java compiler
- The bytecode is finally interpreted by the java virtual machine

Explain why high-level languages might be partially compiled and partially interpreted?

- Partially Compiled programs can be used on different platforms as they are interpreted when run
- Code is optimized for the CPU as machine code is generated at run time
- Source code does not need recompiling so more efficient to run

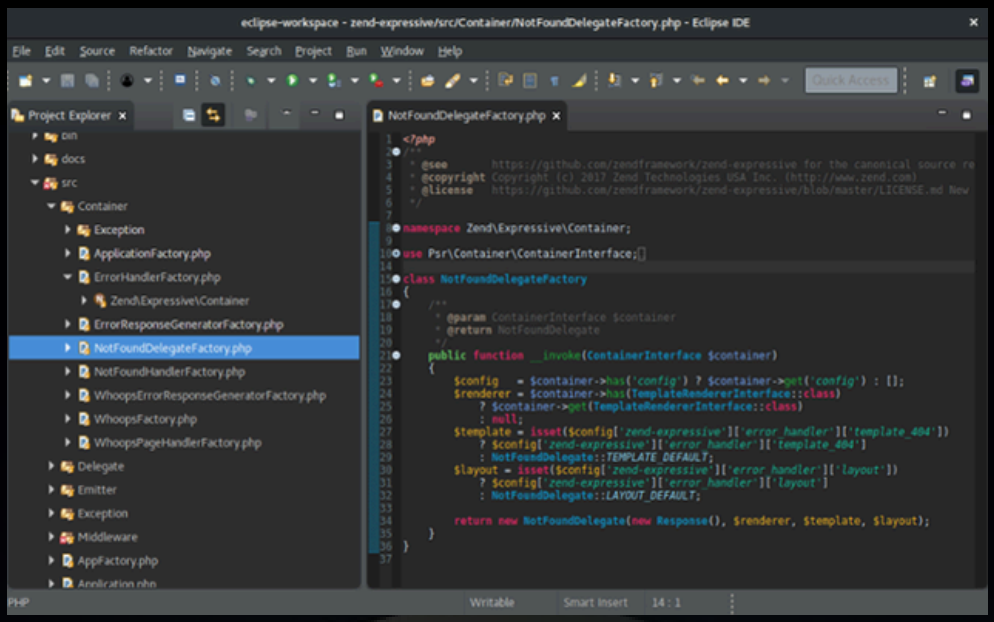
## Assembler

Programs written in assembly language are translated into machine code by an assembler program. Each instruction in the source code consists of an opcode and an operand. The software translates low-level language into machine code for the processor to execute. The source code uses instructions from the processor's instruction set.

## Summary

Compilers are usually used when a high-level language program is complete. They translate all the code at the same time and then run the program. They produce executable/.exe/object code files that can be run without the source code.
Interpreters translate one line of a high-level language program at a time, and then run that line of code. They are most useful while developing the programs because errors can be corrected and then the program continues from that line.
Assemblers are used to translate assembly code into binary/machine code.

# Integrated Development Environment (IDE)




Integrated development Environment is a software application that combines all of the features and tools needed by a software developer

## Features Of An IDE

### 1) Pretty Printing

The pretty print feature in an IDE enhances code readability by applying color-coding which helps in identifying key terms

### 2) Context Sensitive Prompts

Displays predictions of the code being entered and helps to complete statements

### 3) Expand and Collapse Code Blocks

The Expand and Collapse Code Blocks feature in an IDE allows developers to manage code visibility by hiding or showing specific sections.

### 4) Dynamic Syntax checking

Underlines or highlights statements that do not meet the rules of the language and colors syntax errors as the code is entered

### 5) Single Stepping

Executes one line of the program and then stops and runs the code line by line

### 6) Breakpoints

To stop the code at a specific point to the check the current progress

### 7) Report Window

Outputs the contents of the variables and data structure

### 8) Variable Watch

Checks the content of variables at specific points

## Identify features in writing of the program?

- pretty printing
- Auto-complete
- Auto-correct
- Context sensitive prompts
- Expand and collapse code blocks

## Identify debugging tools that IDE can provide?

- Breakpoints
- Single stepping
- Report windows

## Explain how a programmer can make use of a typical Integrated Development Environment (IDE) when writing and testing a program.?

Writing e.g.

- Enter code into an editor
- Pretty printing to identify key terms
- Context-sensitive prompts to help complete statements
- Expand and collapse code blocks
- Auto-complete to suggest what to type next
- Auto-formatting to indent code blocks
- Dynamic syntax checking

Testing e.g.

- Single stepping to run the code line by line
- Breakpoints to stop the code at set points to check values
- Report window to see how variables change

## What are the features provided by an IDE that assist in “Initial Error Detection”

Dynamic Syntax Checking
Identification of Unused Variables
Type Checking

# System Software

## Question 1

**2** Bingwen's computer comes with an Operating System and utility software.

**(a)** Draw **one** line from each utility software to its correct description.

| Utility software | Description |
|---|---|
| | Scans software for errors and repairs the problems |
| Disk formatter | Moves parts of files so that each file is contiguous in memory |
| Defragmentation | Creates a copy of data that is no longer required |
| Back-up | Sets up a disk so it is ready to store files |
| Disk repair | Scans for errors in a disk and corrects them |
| | Creates a copy of data in case the original is lost |

[4]

**(b)** Identify **four** key management tasks that the Operating System will perform.

1 .....

2 .....

3 .....

4 .....

[4]

## Question 2

7 Jennifer is writing a computer program for her A Level homework.

**(a)** Jennifer uses a program library to help her write her computer program.

Describe how a program library can be used while writing a computer program.

.....

.....

.....

..... [2]

**(b)** Jennifer uses an Integrated Development Environment (IDE) to write her computer program.

**(i)** The IDE allows Jennifer to use both an interpreter and a compiler while creating her computer program.

Describe the ways in which Jennifer can use **both** a compiler **and** an interpreter while developing the program.

.....

.....

.....

.....

.....

.....

.....

..... [4]

**(ii)** Identify **two** debugging tools that a typical IDE can provide.

1 .....

.....

2 .....

.....

[2]

## Question 3

**(ii)** A typical IDE provides debugging tools to support the testing of a program.

Identify **three** other tools or features found in a typical IDE to support the writing of the program.

1 .....

2 .....

3 .....

[3]

**(d)** Francis's team use language translators.

Complete the descriptions of language translators by writing the missing words.

..... are usually used when a high-level language program is complete. They translate all the code at the same time and then run the program.

They produce ..... files that can be run without the source code.

..... translate one line of a high-level language program at a time, and then run that line of code. They are most useful while developing the programs because errors can be corrected and then the program continues from that line.

Assemblers are used to translate assembly code into ..... .

[4]

## Question 4

**(b)** Describe the file management tasks that an Operating System performs.

.....

.....

.....

.....

.....

.....

.....

..... [4]

**(c)** Identify **two** utility programs that can be used to improve the performance of a computer **and** state how they improve the performance.

1 .....

.....

.....

.....

2 .....

.....

.....

.....

[4]

## Question 5

5 A programmer uses an Integrated Development Environment (IDE) to develop a program.

**(a)** Draw **one** line from each IDE feature to its correct description.

| IDE feature | Description |
|---|---|
| Context-sensitive prompt | Executes one line of the program and then stops |
| Dynamic syntax check | Underlines or highlights statements that do not meet the rules of the language |
| Breakpoint | Outputs the contents of variables and data structures |
| Single stepping | Stops the code executing at a set line |
| Report window | Displays predictions of the code being entered |

[4]

**(c)** Explain the benefits to the programmer of using program libraries.

.....

.....

.....

.....

.....

.....

.....

..... [3]

## Question 6

6 A programmer uses language translators when writing and testing a program.

**(a)** Describe the operation of a compiler.

.....

.....

.....

..... [2]

**(b)** Describe the operation of an interpreter.

.....

.....

.....

..... [2]

**(c)** Explain how a programmer can make use of a typical Integrated Development Environment (IDE) when writing **and** testing a program.

Writing .....

.....

.....

.....

Testing .....

.....

.....

.....

[4]

## Question 7

4 A computer has system software including an operating system.

**(a)** Describe the key management tasks of an operating system.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

**(b)** Utility software is a type of system software.

**(i)** Describe the purpose of back-up software and defragmentation software.

Back-up software .....

.....

.....

.....

Defragmentation software .....

.....

.....

.....

[4]

**(ii)** Give **one other** example of utility software.

..... [1]

## Question 8

7 **(a)** State **two** benefits to a programmer of using Dynamic Link Library (DLL) files.

1 .....

.....

2 .....

.....

[2]

**(b)** Memory management is one of the tasks performed by an Operating System (OS).

Describe the ways in which memory management organises and allocates Random Access Memory (RAM).

.....

.....

.....

..... [2]

## Question 9

**1** **(a)** Draw **one** line from each utility software to its most appropriate purpose.

| Utility software | Purpose |
|---|---|
| virus checker | to reorganise files so they are contiguous |
| disk formatter | to scan for malicious program code |
| backup | to decrease the file size |
| disk repair | to initialise a disk |
| defragmentation | to create copies of files in case the original is lost |
| | to check for and fix inconsistencies on a disk |

[5]

**(b)** Compilers and interpreters translate programs written in a high-level language into a low-level language.

**(i)** State **two** drawbacks of using a compiler compared to an interpreter during program development.

1 .....

.....

.....

2 .....

.....

.....

[2]

**(ii)** Explain why high-level language programs might be partially compiled and partially interpreted.

.....

.....

.....

..... [2]

## Question 10

**3** Draw **one** line from each Operating System (OS) management task to its most appropriate description.

| OS Management task | Description |
| --- | --- |
| hardware management | dynamically allocates memory to processes |
| security management | marks unallocated file storage for availability |
| memory management | installs programs for devices connected to external ports |
| process management | validates user and process authenticity |
| | allows processes to transfer data to and from each other |

[4]

## Question 11

3 A computer has an Operating System (OS).

**(a)** Describe how the Operating System manages the peripheral hardware devices of the computer.

.....

.....

.....

.....

.....

.....

.....

..... [4]

**(b)** Hardware management is one key management task carried out by the Operating System.

Identify **two other** key management tasks carried out by the Operating System.

1 .....

2 .....

[2]

**(c)** The Operating System has utility software including defragmentation software.

Explain how defragmentation can improve the performance of the computer.

.....

.....

.....

.....

.....

..... [3]

## Question 12

**(d)** The laptop has systems software.

**(i)** Describe how the Operating System (OS) manages processes in the computer.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

**(ii)** Describe the purpose of utility software in a computer.

.....

.....

.....

..... [2]

## Question 13

7 A software developer is working in a team writing a program for a client.

**(a)** The developer is writing a new program library to be used by the other team members.

**(i)** Define the term **program library**.

.....

.....

.....

..... [2]

**(ii)** Explain **two** benefits to the developer of choosing to create a Dynamic Link Library (DLL).

1 .....

.....

.....

.....

2 .....

.....

.....

.....

[4]

**(b)** The development team needs to use a translator whilst writing the program for the client.

Identify whether an interpreter or a compiler would be more appropriate at this stage of the program development.

Justify your choice.

Translator .....

Justification .....

.....

.....

.....

.....

..... [3]

**(c)** The development team uses an Integrated Development Environment (IDE).

Complete the table by describing the typical features found in an IDE.

| Feature | Description |
|---|---|
| Breakpoints | ..... ..... ..... |
| Dynamic syntax checks | ..... ..... ..... |
| Context-sensitive prompts | ..... ..... ..... |
| Single stepping | ..... ..... ..... |

[4]

## Question 14

5 A programmer is developing a computer game in a high-level language to sell to the public.

**(a)** The programmer uses both an interpreter and a compiler at different stages of the development of the program.

**(i)** Explain the reasons why the programmer uses an interpreter while writing the program code.

.....

.....

.....

..... [2]

**(ii)** Explain the reasons why the programmer uses a compiler when the program has been written.

.....

.....

.....

.....

.....

..... [3]

## Question 15

6 A programmer uses both a compiler and an interpreter to translate a program written in a high-level language.

**(a)** Describe the advantages of using the interpreter compared to the compiler to translate the program.

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

**(b)** State **one** reason why some high-level languages are partially compiled and partially interpreted.

.....

..... [1]

**(c) (i)** Identify **two** features that support the visual presentation of the code in a typical Integrated Development Environment (IDE).

1 .....

.....

2 .....

.....

[2]

**(ii)** Identify **two** features that support the debugging of the code in a typical IDE.

1 .....

.....

2 .....

.....

[2]

## Question 16

8 A computer has an Operating System (OS).

**(a)** State **one** purpose of the Operating System.

.....

..... [1]

**(b)** The Operating System has utility software.

The purpose of some utility software is to improve security.

Identify **one** example of utility software that is **not** intended to improve security.

Explain why this software is needed.

Utility software .....

Explanation .....

.....

.....

.....

.....

[3]

**(c)** An optical disc reader/writer is connected to the computer.

**(i)** Give the name of **one** port that can provide a connection for the optical disc reader/writer.

..... [1]

**(ii)** Describe the roles of the address bus, the data bus **and** buffers in the process of writing data to the optical disc reader/writer.

.....

.....

.....

.....

.....

..... [3]

## Question 17

5 A programmer uses an Integrated Development Environment (IDE) to develop a program that monitors air quality.

**(a)** Describe the following features of a typical IDE.

Context-sensitive prompts .....

.....

.....

.....

Single stepping .....

.....

.....

.....

.....

[4]

**(ii)** The program files are stored on a new hard disk after they have been downloaded.

Describe the reasons why a hard disk formatter is needed for the new hard disk.

.....

.....

.....

.....

.....

..... [3]

## Question 18

3 A software developer is writing a computer program.

**(a)** The developer uses an interpreter while writing the program code because it is easier for debugging.

Explain **one** reason why it is easier to debug the program code using an interpreter instead of a compiler.

.....

.....

.....

..... [2]

**(b)** The program is ready to be sold to customers.

The developer uses a compiler because it creates an executable file.

Explain the reasons why the need to create an executable file makes the complier the appropriate choice when the program is complete.

.....

.....

.....

.....

.....

..... [3]

## Question 19

6 A computer has an Operating System (OS).

Memory management and process management are two OS tasks.

Explain how memory management **and** process management support multi-tasking.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

## Question 20

8 A programmer uses an Integrated Development Environment (IDE) to write a computer program. The IDE has both a compiler and an interpreter as built-in translators.

**(a)** The programmer decides to use the compiler when testing the final program.

Describe the benefits of using the compiler during testing.

.....

.....

.....

..... [2]

**(b)** IDEs have many features other than built-in translators.

Complete the table by identifying **one other** common IDE feature that can be used for each purpose. Describe how each feature helps the user during program development.

Each feature must be different. Do **not** give translator as one of your features.

| Purpose | IDE feature | Description |
|---|---|---|
| for coding | ..... ..... | ..... ..... ..... ..... |
| for presentation | ..... ..... | ..... ..... ..... ..... |
| for debugging | ..... ..... | ..... ..... ..... ..... |

[6]

**(c)** The programmer uses program libraries when developing the program.

Describe **two** benefits to the programmer of using program libraries.

1 .....

.....

2 .....

.....

[2]

## Question 21

**(e)** Program libraries were used when writing the robot's software.

**(i)** State what is meant by a **program library**.

.....

..... [1]

**(ii)** Some program libraries include Dynamic Link Library (DLL) files.

Describe the benefits of a programmer using a library with DLL files instead of using a library that does not include DLL files.

.....

.....

.....

.....

.....

.....

.....

..... [4]

## Question 22

Bart plays computer games on his stand-alone games console. The games console has an operating system.

(a) Describe the tasks performed by the operating system to manage the main memory in the games console.

.....

.....

.....

.....

.....

.....

.....

..... [4]

(b) The computer games are written in a high-level language. Bart does not need a compiler or an interpreter to run the games he buys for his console. Explain why the games run without the need for a compiler or an interpreter.

.....

.....

.....

..... [2]

## Question 23

9 Utility programs are examples of system software.

**(a)** Complete the table by writing the name of the utility program for each description.

| Description | Utility program |
|---|---|
| Reorganises files on a disk to improve efficiency | |
| Scans a hard disk to identify bad sectors | |
| Prepares a hard disk for first use | |

[3]

## Question 24

(c) Amir's computer has system software, including utility software and an operating system. (i) Explain how the disk formatter, disk contents analysis and disk repair utilities work together.

.....

.....

.....

.....

.....

..... [3]

(ii) Amir's computer has several peripheral devices connected to it. State three peripheral management tasks performed by the operating system.

Task 1 .....

.....

Task 2 .....

.....

Task 3 .....

..... [3]

## Question 25

4 Annchi is writing a computer game with a group of friends. (a) One of her friends has suggested using Dynamic Link Library (DLL) files to help them develop the game. (i) Give three reasons why Annchi and her friends should use DLL files when developing the game.

1 .....

.....

2 .....

.....

3 .....

..... [3]

(ii) Give two reasons why Annchi and her friends should not use DLL files when developing the game.

1 .....

.....

2 .....

..... [2]

(b) Each member of the group is creating a different part of the game. Each person needs to test their part of the game independently before they are combined. Identify the most appropriate type of translator that should be used to test each part of the game independently. Justify your choice.

Translator .....

Justification .....

.....

.....

.....

..... [3]

## Question 26

3 Kimmy has written a program in a high-level language. (a) Kimmy has used library routines in the program.

(i) Describe two advantages of using library routines in the program.

1 .....

.....

.....

.....

2 .....

.....

.....

.....[4]

(ii) Describe what is meant by a Dynamic Link Library (DLL).

.....

.....

.....

..... [2]

(b) Three translators are compilers, interpreters, and assemblers.

(i) State one benefit of Kimmy using an interpreter during the development of the program.

.....

..... [1]

(ii) State three benefits of Kimmy using a compiler when the program is complete.

1 .....

.....

2 .....

.....

3 .....

..... [3

## Question 27

A computer has an operating system (OS) and utility software.

**(a)** The following table lists key management tasks performed by an operating system and their descriptions.

Complete the table by writing the missing management task names and descriptions.

| Management task | Description |
|---|---|
| Memory management | |
| | Provides user accounts and passwords |
| | Handles the signals sent when the attention of the processor is required elsewhere |
| Provision of a software platform | |

[4]

(b) A hard disk formatter and a hard disk defragmenter are two examples of utility software.

(i) Describe the actions performed by a hard disk formatter and a hard disk defragmenter.

Hard disk formatter .....

.....

.....

. .....

Hard disk defragmenter .....

.....

.....

.....

..... [4

(ii) Identify three other examples of utility software that can be installed on the computer.

1 .....

.....

2 .....

.....

3 .....

..... [3]

## Question 28

(b) Aaron's computer has an operating system (OS). The OS manages the running processes and provides a user interface. Describe these OS management tasks.

Process management .....

.....

.....

.....

.....

.....

Provision of a user interface .....

.....

.....

.....

.....

.....

[6]

(c) Aaron's computer has a virus checker and backup software.

Describe these utility programs.

Virus checker .....

.....

.....

.....

Backup software .....

.....

.....

..... [4]

(d) Aaron creates a web page using JavaScript code and HTML tags.

Describe how the JavaScript code is translated using an interpreter.

.....

.....

.....

..... [2]

## Question 29

**1 (a)** The diagram shows different types of software on the left, and descriptions on the right.

Draw a line from each type of software to its correct description.

| Type of software | Description |
| --- | --- |
| Operating system | Provides a ready-built routine that can be imported into a program |
| Utility program | Provides an interface between the user and the hardware |
| Library program | Converts source code into a low-level language |
| Compiler | Creates a new document for the user to edit |
| | An additional program that helps to maintain or configure the system |

[4]

b) Describe the purpose of disk repair software.

.....

.....

.....

.....

.....

.....[3

## Question 30

2 Leonardo's mobile phone has an operating system (OS).

(a) Describe the following key management tasks that the mobile phone operating system carries out.

Process management .....

.....

.....

.....

. .....

.....

Memory management .....

.....

.....

.....

.....

..... [6]

## Question 31

An operating system (OS) is usually pre-installed on a new computer.

(a) The OS performs a number of different tasks such as memory management and security management.

(i) State three memory management tasks the OS performs.

1 .....

.....

2 .....

.....

3 .....

..... [3]

(ii) State three security management tasks the OS performs.

1 .....

.....

2 .....

.....

3 .....

..... [3]

(iii) State two tasks, other than memory management and security management that are carried out by an OS.

1 .....

.....

2 .....

..... [2]

**(b)** Utility software is usually pre-installed on a new computer.

The following table lists four programs. Put **one** tick (✓) in each row to indicate whether or not the program is utility software.

| Program | True | False |
|---|---|---|
| Disk Defragmenter | | |
| Word Processor | | |
| Library Program | | |
| Compression Software | | |

[4]

## Question 32

An operating system (OS) is usually pre-installed on a new computer.

(a) The OS performs a number of different tasks such as file management and peripheral management.

(i) State three file management tasks the OS performs.

1 .....

.....

2 .....

.....

3 .....

..... [3]

(ii) State three printer management tasks the OS performs.

1 .....

.....

2 .....

.....

3 .....

..... [3]

**(b)** Utility software is usually pre-installed on a new computer.

**(i)** The following table lists four programs. Put **one** tick (✓) in each row to indicate whether or not the program is utility software.

| Program | True | False |
|---|---|---|
| Database | | |
| Virus checker | | |
| Web browser | | |
| Backup software | | |

[4]

**(ii)** Name **two** other utility programs.

Program 1 .....

Program 2 .....

[2]

## Question 33

(b) Kim will use library routines in her program.

(i) Describe what is meant by a library routine.

.....

.....

.....

.....[2]

ii) Describe one benefit and one drawback of using library routines.

Benefit .....

.....

.....

.....

Drawback .....

.....

.....

.....[4]

(c) Kim develops her program and makes it ready for use. To do this, she uses first an interpreter and then a compiler. Explain why Kim needs to use both an interpreter and a compiler.

Interpreter .....

.....

.....

.....

Compiler .....

.....

.....

..... [4]

## Question 34

Draw a line to link each OS management task to the appropriate user action.

| OS management task | Action |
| --- | --- |
| Main memory management | The user moves the mouse on the desktop |
| Input/Output management | The user closes the spreadsheet program |
| Secondary storage management | The user selects the Save command to save their spreadsheet file |
| Human computer interface management | The user selects the Print command to output their spreadsheet document |

[3]

(b) A user has the following issues with the use of his PC.

State the utility software which should provide a solution.

(i) The hard disk stores a large number of video files. The computer frequently runs out of storage space. Utility software solution .....[1]

(ii) The user is unable to find an important document. He thinks it was deleted in error some weeks ago. This must not happen again. Utility software solution

.....[1]

(iii) The operating system reports 'Bad sector' errors. Utility software solution

.....[1]

(iv) There have been some unexplained images and advertisements appearing on the screen. The user suspects it is malware. Utility software solution

.....[1]

## Question 35

(a) (i) Explain why a personal computer (PC) needs an operating system (OS).

.....

.....

.....[2]

(ii) One of the tasks carried out by the OS is the management of the use of the processor.

Name and describe two other management tasks that the OS performs.

1 .....

.....

.....

2 .....

.....

..... [4

(b) A user has the following issues with the use of their personal computer (PC).

For each case, state the utility software which should provide a solution.

(i) The user wants to send a large file as an attachment to an email. The user knows that the recipient's Internet Service Provider (ISP) has a limit of 2MB for file attachments. Utility software solution: .....[1]

(ii) The user is writing a book and is worried that the document files could get damaged or deleted. Utility software solution: .....[1]

(iii) The computer has recently been slow to load large files. The user has deleted a large number of small files to try to solve the problem. A friend has advised that there is a procedure which should be regularly carried out to reorganise file storage on the hard disk. Utility software solution: .....[1]

(iv) The user clicked on an attachment in an unsolicited email. Since then, the computer has shown some unexplained behaviours. Utility software solution: .....[1]

## Question 36

2 **(a)** The diagram shows three items of software that translate program code.

Draw **one** line from each context to the correct item of translation software.

| Context | Item of translation software |
| --- | --- |
| A web page contains a client-side script. | Assembler |
| Each instruction in the source code consists of an op code and an operand. | Interpreter |
| The source code is required at run-time. | Compiler |
| When the source code is translated, copies of the executable program can be distributed without the need for the source code. | |

[4]

(b) The Java programming language is said to be machine or platform independent.

(i) Describe what is meant by machine independent.

.....

.....[1]

(ii) Describe how a Java source code program is translated.

.....

.....

.....

.....[2]

## Question 37

Three examples of language translators and four definitions are shown below.

Draw lines to link each language translator to the correct one or more definitions.

| Language translator | Definition |
| --- | --- |
| Compiler | The software reads the source code and reports all errors. The software produces an executable file. |
| Assembler | The software reads each statement and checks it before running it. The software halts when it encounters a syntax error. |
| Interpreter | The software translates a high-level language program into machine code for the processor to execute. |
| | The software translates low-level statements into machine code for the processor to execute. |

[3]

## Question 38

1 Describe two differences between a compiler and interpreter.

1 .....

.....

.....

.....

2 .....

.....

.....

.....[4]

## Question 39

(b) (i) Explain why a computer needs an operating system.

.....

.....

.....

.....[2]

(ii) Give two key management tasks carried out by an operating system.

1 .....

.....

2 .....

.....[2]

(c) New program code is to be written in a high-level language. The use of Dynamic Link Library (DLL) files is considered in the design.

Describe what is meant by a DLL file.

.....

.....

.....

.....[2]

## Question 40

A small company produces scientific magazines. The owner buys some new desktop computers. The computers are used to store thousands of colour images (diagrams and photographs). All the computers have Internet access.

(a) Name three utility programs the company would use on all their computers. Describe what each program does.

1 .....

Description .....

.....

2 .....

Description .....

.....

3 .....

Description .....

.....[6]

## Question 41

One management task carried out by an operating system is to provide a user interface.

Describe two more of these management tasks.

1 .....

.....

.....

.....

2 .....

.....

.....

.....[4]

## Question 42

A programmer is writing a program that includes code from a program library.

(a) Describe two benefits to the programmer of using one or more library routines.

1 .....

.....

.....

.....

2 .....

.....

.....

.....[4]

(b) The programmer decides to use a Dynamic Link Library (DLL) file.

(i) Describe two benefits of using DLL files.

1 .....

.....

.....

.....

2 .....

.....

.....

.....[4]

(ii) State one drawback of using DLL files.

.....

.....

.....[2]

## Question 43

**2** Assemblers translate from assembly language to machine code. Some assemblers scan the assembly language program twice; these are referred to as two-pass assemblers.

The following table shows five activities performed by two-pass assemblers.

Write 1 **or** 2 to indicate whether the activity is carried out during the first pass or during the second pass.

| Activity | First pass or second pass |
|---|---|
| any symbolic address is replaced by an absolute address | |
| any directives are acted upon | |
| any symbolic address is added to the symbolic address table | |
| data items are converted into their binary equivalent | |
| forward references are resolved | |

[5]

## Question 44

**11** A game program is written which can be either interpreted or compiled. The table below shows five statements about the use of interpreters and compilers.

Tick (✓) to show whether the statement refers to an interpreter or to a compiler.

| Statement | Interpreter | Compiler |
|---|---|---|
| This translator creates an executable file | | |
| When this translator encounters a syntax error, game execution halts | | |
| The translator analyses and checks each line just before executing it | | |
| This translator will produce faster execution of the game program | | |
| Use of this translator makes it more difficult for the user to modify the code of the game | | |

[5]

## Question 45

**(a)** Describe the difference between a command line interface (CLI) and a graphical user interface (GUI).

.....

.....

.....

..... [2]

**(b)** CLI and GUI interfaces have advantages to certain users.
Describe which type of user would find each of the interfaces the most useful. Justify your choice.

**CLI** .....

.....

.....

**GUI** .....

.....

..... [2]

## Question 46

**(d)** The computer is running a single-user operating system.

Describe what this means.

.....

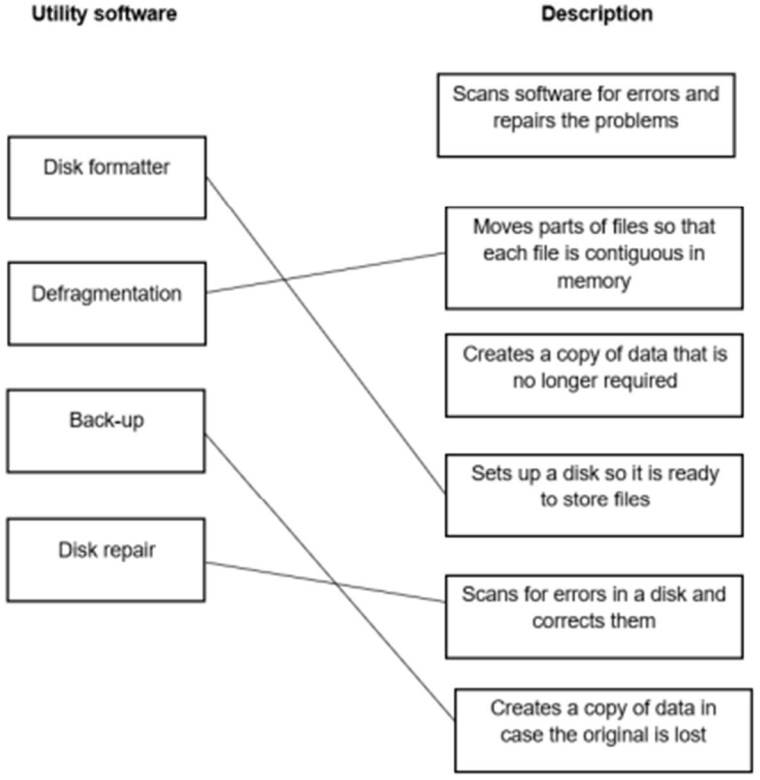
.....

.....

..... [2]

# Answer

## Answer 1

| | | |
|---|---|---|
| 2(a) | **1 mark** for each correct line<br>  | **4** |
| 2(b) | **1 mark** per bullet point to **max 4**<br>• memory management<br>• file management<br>• security management<br>• hardware / device / peripheral / resources management<br>• input/output management<br>• process management<br>• error checking and recovery<br>• provision of a platform for software<br>• provision of a user interface | **4** |

## Answer 2

| | | |
|---|---|---|
| 7(a) | **1 mark** per bullet point to **max 2**<br><br>• Program libraries store pre-written functions and routines<br>• The program library can be referenced/imported<br>• the functions/routines can be called in her own program | **2** |
| 7(b)(i) | **1 mark** per bullet point to **max 4; max 3** from each section<br><br>Interpreter:<br>• Use an interpreter while writing the program<br>• … to test/debug the partially completed program<br>• … because errors can be corrected and processing continue from where the execution stopped // errors can be corrected in real time // errors are identified one at a time<br><br>Compiler:<br>• Use the compiler after the program is complete<br>• … to create an executable file<br>• Use the compiler to repeatedly test the same (completed) section<br>• … without having to re-interpret every time // compiler not needed at run-time | **4** |
| 7(b)(ii) | **1 mark** per correct tool to **max 2**<br><br>e.g.<br>• Breakpoints<br>• Single stepping<br>• Report windows | **2** |

## Answer 3

| | | |
|---|---|---|
| 4(b)(ii) | **1 mark** for each correct tool<br>e.g.<br>• Colour coding // pretty printing<br>• Auto-complete<br>• Auto-correct<br>• Context sensitive prompts<br>• Expand and collapse code blocks | **3** |

| | | |
|---|---|---|
| 4(d) | **1 mark** for each correctly completed term<br><br>**Compilers** are usually used when a high-level language program is complete. They translate all the code at the same time and then run the program. They produce **executable/.exe/object code** files that can be run without the source code.<br>**Interpreters** translate one line of a high-level language program at a time, and then run that line of code. They are most useful while developing the programs because errors can be corrected and then the program continues from that line.<br>Assemblers are used to translate assembly code into **binary/machine code**. | **4** |

## Answer 4

| | | |
|---|---|---|
| 7(b) | **1 mark** for each bullet point to **max 4**<br><br>• Storage space is divided into file allocation units<br>• Space is allocated to particular files<br>• Maintains / creates directory structures<br>• Specifies the logical method of file storage (e.g. FAT or NTFS)<br>• Provides file naming conventions<br>• Controls access // implements access rights // implements password protection // Makes file sharing possible<br>• Specifies tasks that can be performed on a file (e.g. open, close, delete, copy, create, move etc.) | **4** |
| 7(c) | **1 mark** for identifying program **1 mark** for description, **max 2** per program<br>e.g.<br>• Defragmentation<br>• Less time is taken to access files because each one is contiguous so there is less head movement<br><br>• Virus checker<br>• makes more RAM available for programs to run<br>• ... because it removes software that might be taking up memory / replicating<br><br>• Disk repair / Disk contents analysis<br>• preventing bad sectors being used because it identifies / marks them<br>• reduces access times by optimising storage<br><br>• Disk/system clean up<br>• releases storage by removing unwanted / temporary files | **4** |

## Answer 5

| | | |
|---|---|---|
| 5(a) | **1 mark** for 1 correct line, **2 marks** for 2 correct lines, **3 marks** for 3 or 4 correct lines, **4 marks** for all 5 correct lines<br><br>**IDE feature** / **Description**<br><br>Context-sensitive prompt → Executes one line of the program and then stops<br>Dynamic syntax check → Underlines or highlights statements that do not meet the rules of the language<br>Breakpoint → Stops the code executing at a set line<br>Single stepping → Outputs the contents of variables and data structures<br>Report window → Displays predictions of the code being entered | **4** |
| 5(c) | **1 mark** per bullet point to **max 3**<br><br>• Saves (programming/testing) time as code does not have to be written/re-written from scratch // code does not have to be tested<br>• Code is already tested so it is more robust/likely to work<br>• If there is an improvement in the library routine the program updates automatically<br>• can perform complex calculations that the programmer may be unable to do | **3** |

## Answer 6

| | | |
|---|---|---|
| 6(a) | **1 mark** per point to **max 2**<br>e.g.<br>• Attempts to **translate** the whole source code<br>• Creates a separate error report at the end of the translation process<br>• If translation successful / no errors creates an **executable** file | **2** |
| 6(b) | **1 mark** per point to **max 2**<br>e.g.<br>• Reads each line then translates it **and executes** it<br>• Stops when an error is encountered // displays errors where it finds them | **2** |
| 6(c) | **1 mark** per point, **max 2** for writing, **max 2** for testing<br><br>Writing e.g.<br>• Enter code into an editor<br>• Pretty printing to identify key terms<br>• Context-sensitive prompts to help complete statements<br>• Expand and collapse code blocks<br>• Auto-complete to suggest what to type next<br>• Auto-formatting to indent code blocks<br>• Dynamic syntax checking<br><br>Testing e.g.<br>• Single stepping to run the code line by line<br>• Breakpoints to stop the code at set points to check values<br>• Report window to see how variables change | **4** |

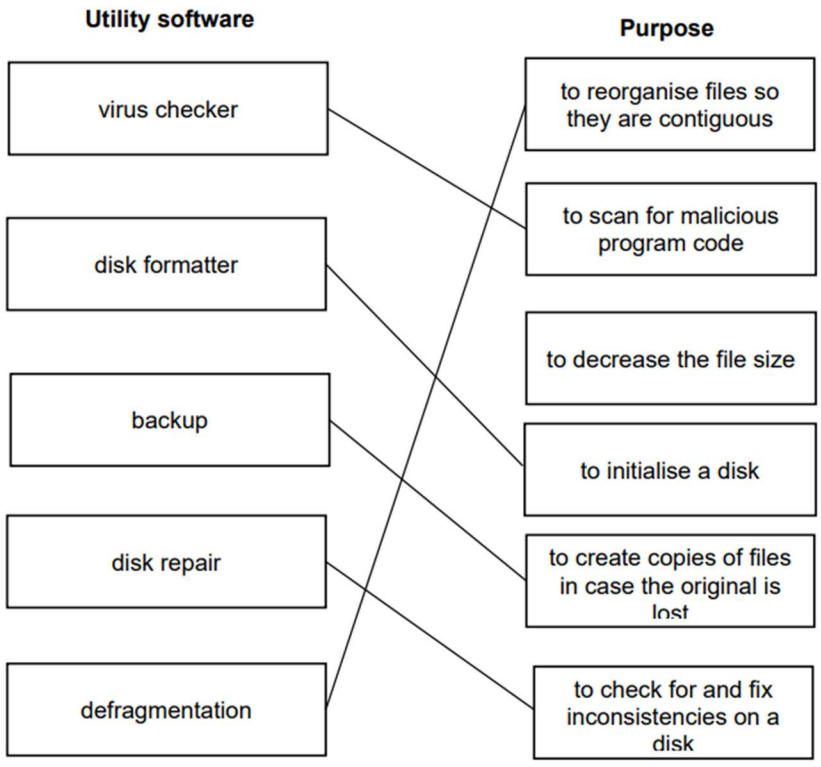
## Answer 7

| | | |
|---|---|---|
| 4(a) | **1 mark** for identifying task, **max 2** for each description<br>**Max 2** for only identifying tasks without descriptions<br><br>e.g.<br>• Memory management<br>• Controls the movement of data between RAM, processor, VM etc<br>• allocates memory to processes<br><br>• File management<br>• Creates files/folders<br>• Renames file/folders<br><br>• Security management<br>• Creates accounts/passwords<br>• Provide /upgrade firewall / anti-malware<br><br>• Hardware management<br>• Receives data from input devices ///sends data to output device<br>• Use of device drivers<br><br>• Process management<br>• Decides which process to run next<br>• supports multitasking | **4** |
| 4(b)(i) | **1 mark** per point to **max 2** for each<br><br>Back-up<br>• To make a copy of data **at regular intervals**<br>• So that if it is lost/corrupted it **can be retrieved**<br><br>Defragmentation<br>• Make individual **files** occupy contiguous blocks // move free space together<br>• Improve disk access times // Data/files can be loaded faster | **4** |
| 4(b)(ii) | **1 mark** from<br>e.g.<br>• Compression software<br>• (Hard) disk formatter<br>• Virus checker<br>• Disk analysis software<br>• Disk repair software | **1** |

## Answer 8

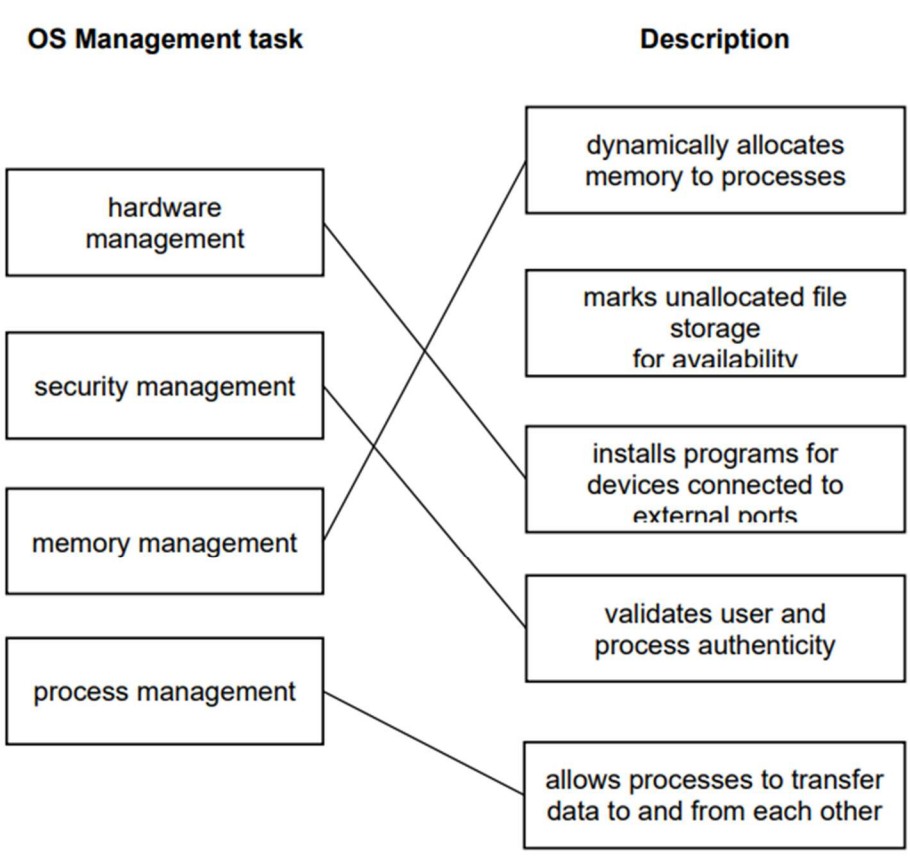
| | | |
|---|---|---|
| 7(a) | **1 mark** for each benefit (**max 2**):<br><br>• (main) memory requirements for program are reduced as dynamic link library is loaded only once / when required<br>• the executable file size is smaller because the executable does not contain all the library routines<br>• maintenance not needed to be done by the programmer because the DLL is separate from program<br>• no need to recompile the main program when changes are made to DLL because changes / improvements/ error correction to the DLL file code are done independently of the main program | **2** |
| 7(b) | **1 mark** for each bullet point (**max 2**):<br><br>• RAM is assigned into blocks<br>• dynamic allocation of RAM to programs / processes<br>• reclaims unused blocks of RAM<br>• prevents two programs / processes occupying the same area of RAM at the same time<br>• moves data from secondary storage when needed // manages paging, segmentation and virtual memory | **2** |

## Answer 9

| | | |
|---|---|---|
| 1(a) | **1 mark** for each correct line.<br><br>  <br> | **5** |

| | | |
|---|---|---|
| 1(b)(i) | **1 mark** for each bullet point **(max 2)**:<br>• larger amounts of source code take time to compile<br>• slower to produce the object code than an interpreter<br>• code cannot be changed without recompilation<br>• the program will not run if there are any errors<br>• errors cannot be corrected in real-time<br>• one error may result in other false errors being reported<br>• cannot easily test specific sections of the source code // cannot easily test unfinished source code | **2** |
| 1(b)(ii) | **1 mark** for each bullet point **(max 2)**:<br>• partially compiled programs can be used on different platforms as they are interpreted when run<br>• code is optimised for the CPU as machine code is generated at run time | **2** |

## Answer 10

| | | |
|---|---|---|
| 3 | **1 mark** for each correct line:<br>  | **4** |

## Answer 11

| | | |
|---|---|---|
| 3(a) | **1 mark** each to **max 4**<br><br>Examples:<br>• **Installs** device drivers<br>• … to allow communication between peripherals and computer<br>• Sends data and receives data to and from peripherals<br>• … such as to an output device and from an input device/by example<br>• Handles buffers for transfer of data<br>• … to ensure smooth transfer between devices that transmit and receive at different speeds<br>• Manages interrupts / signals from the device | **4** |
| 3(b) | **1 mark** each to **max 2**<br><br>• Memory management<br>• File management<br>• Security management<br>• Process management<br>• Error checking and recovery | **2** |
| 3(c) | **1 mark** each to **max 3**<br><br>• Rearranges blocks of individual files (on the HDD) so they are contiguous // moves the free space together<br>• Accessing each file is faster<br>• …because there is no need to search for the next fragment / block of the file<br>• …so less head movement is needed | **3** |

## Answer 12

| | | |
|---|---|---|
| 5(d)(i) | **1 mark** each to **max 5**<br><br>• Manages the scheduling of processes // decides which order to run processes<br>• Manages which resources the processes require<br>• … such as allocating memory<br>• Enables processes to share data<br>• Prevents interference between processes // resolution of conflicts<br>• Handles the process queue<br>• It allows multi-tasking / multi-processing<br>• ... by ensuring fair access, handling priorities **and** handling interrupts | **5** |
| 5(d)(ii) | **1 mark** each to **max 2**<br><br>• To help users to set-up / configure / analyse / optimise / maintain the computer ...<br>• ... by for example, making memory allocation more efficient<br>• ... by for example, checking the system for faults | **2** |

## Answer 13

| | | |
|---|---|---|
| 7(a)(i) | **1 mark** each to max 2<br><br>• Set of pre-written / pre-compiled / pre-tested subroutines<br>• ... which can be **called** in other programs<br>• ... by **installing/importing** the library | **2** |
| 7(a)(ii) | **1 mark** for each bullet point. Mark in pairs; **1 mark** for a benefit and **1 mark** for an appropriate expansion<br><br>• (main) memory requirements for program is reduced<br>• ... as dynamic link library is loaded only once / when required<br><br>• the executable file size of the program using the DLL will be smaller<br>• ... because the executable does not contain (all) the library routines<br><br>• maintenance not needed to be done by the programmer<br>• ... because the DLL is separate from program<br><br>• no need to **recompile** the main program when changes are made to DLL<br>• ... because changes / improvements/ error correction to the DLL file code are done independently of the main program<br><br>• A single DLL file can be made available to several application programs<br>• ... Saving space in memory / easing the pressure on memory | **4** |

| | | |
|---|---|---|
| 7(b) | No mark for choice. **1 mark** each to **max 3** for justification<br><br>Interpreter<br>• Allows the developer to make real-time changes<br>• … so the program can be debugged **at each stage**<br>• ... the effect of any changes made by the developer can be seen immediately<br>• The developer can test when incomplete<br>• … so small parts can be tested without having to test the rest of the program<br>• … if one section does not work others can still be tested<br>• To avoid dependent errors<br><br>Compiler<br>• The developer can debug multiple errors simultaneously<br>• Produces an executable file<br>• … so that the developer can test the program multiple times without recompiling | **3** |
| 7(c) | **1 mark** each<br><br>Breakpoints:<br>• **Stop the code** at a specific line to check the current progress / values<br><br>Dynamic syntax checks:<br>• Highlight / underline / colour syntax errors **as the code is entered**<br><br>Context-sensitive prompts:<br>• Suggest the code to add // automatically complete statements<br><br>Single stepping:<br>• Run the code **one line at a time** so the values can be checked | **4** |

## Answer 14

| | | |
|---|---|---|
| 5(a)(i) | **1 mark** each to **max 2**<br><br>• Programmer can test sections of the code without every part working / being written<br>• Programmer can debug in real time<br>• … so that errors can be fixed and the program continued from that point<br>• The effect of any changes made by the programmer can be seen immediately<br>• To avoid dependent errors | **2** |
| 5(a)(ii) | **1 mark** each to **max 3**<br><br>• The compiler produces an executable file<br>• ... so the user cannot access / edit / sell the code<br>• ... and users do not need the translator to run the game<br>• The game can be compiled for different hardware specifications<br>• ... and then used to generate more income for the programmer<br>• The program can be **tested** multiple times without having to retranslate each time | **3** |

## Answer 15

| | | |
|---|---|---|
| 6(a) | **1 mark** for each bullet point (**max 4**)<br><br>• easier to debug the program<br>• … because it translates line-by-line and stops when an error is found<br>whereas the compiler translates all the program at the same time<br>• … only reporting one error at a time<br>• … which allows the error to be corrected in real time<br>whereas the program would need to be corrected and recompiled<br>• ... and the program can restart at same point when error occurred<br>with a compiler the program needs to be re-run<br>• The effect of any changes made by the programmer can be seen<br>immediately with a compiler the effects can only be seen after re-running<br>• A partially completed program can be translated / tested on its own<br>a compiler cannot translate a partial program | **4** |
| 6(b) | **1 mark** for each bullet point (**max 1**)<br><br>• Partially compiled programs can be used on different platforms as they are interpreted when run<br>• Code is optimised for the CPU as machine code is generated at run time<br>• Source code does not need recompiling so more efficient to run | **1** |

| | | |
|---|---|---|
| 6(c)(i) | **1 mark** for each bullet point (**max 2**)<br><br>• Prettyprint<br>• Expand/collapse code blocks<br>• **Auto** indentation / formatting | **2** |
| 6(c)(ii) | **1 mark** for each bullet point **(max 2)**<br><br>• Single stepping<br>• Breakpoints<br>• Report window<br>• Variable expressions | **2** |

## Answer 16

| | | |
|---|---|---|
| 8(a) | **1 mark** for each bullet point (**max 1**)<br><br>• To hide the complexities of the hardware from the user<br>• To provide a platform for software to run<br>• To provide a user interface | **1** |
| 8(b) | **1 mark** for the name of the utility software<br>**2 marks** for the explanation<br><br>• Defragmentation software<br>• … because over time saving and deleting of small files fragments the disk<br>• ... the software makes (individual) files contiguous<br>• … so access time to the files is improved<br>• … because head movement is reduced<br><br>• Disk contents analysis/disk repair software<br>• … to identify and mark bad sectors<br>• ... to restore corrupted files<br>• ... to recover lost data (due to hardware failure)<br><br>• File compression<br>• … to reduce the size of files<br>• ... which saves storage and memory space // by example<br>• ... and reduces transmission time // by example<br><br>• Disk formatter<br>• … to prepare a disk for use // set up the file system<br>• ... to partition the disc<br>• ... to delete all the data from the disc | **3** |

| | | |
|---|---|---|
| 8(c)(i) | **1 mark** for each bullet point (**max 1**)<br><br>• USB / Universal Serial Bus<br>• HDMI | **1** |
| 8(c)(ii) | **1 mark** for each component (**max 3**)<br><br>Buffers<br>• A buffer temporarily holds data until it is ready to be transmitted **to the device**<br><br>Address Bus<br>• The address of the **data to be written to the device** (in RAM) is carried on the address bus<br><br>Data Bus<br>• All data to be **written to the device / buffer** is carried on the data bus | **3** |

## Answer 17

| | | |
|---|---|---|
| 5(a) | **1 mark** for each bullet point (**max 2** for each feature).<br><br>Context-sensitive prompts:<br>• As the code is being written<br>• …the options to complete the statement are shown<br><br>Single stepping:<br>• allows the programmer to execute the program one line at a time<br>• …so that the effects of each statement can be seen | **4** |
| 5(b)(ii) | **1 mark** for each bullet point (**max 3**).<br><br>• Disk needs to be prepared for initial use<br>• Disk needs to be checked for errors<br>• A new file system needs to be generated on the disk<br>• The file allocation table needs to be set up | **3** |

## Answer 18

| | | |
|---|---|---|
| 3(a) | **1 mark** each to **max 2**:<br><br>• The interpreter will stop when an error is found<br>• ... so the error can be corrected in real-time, and the result of changes seen immediately<br><br>• Only one error is displayed at a time<br>• … so fewer errors to correct simultaneously **and** no dependent errors | **2** |
| 3(b) | **1 mark** each to **max 3**:<br><br>• Program can be distributed without source code<br>• ... so it cannot be edited/stolen/plagiarised<br><br>• Users do not require the translator to run the program<br>• … so time is not spent retranslating by user | **3** |

## Answer 19

| | | |
|---|---|---|
| 6 | **1 mark** each to **max 4**<br><br>**Max 3** marks for each management task:<br><br>*Memory management:* **Max 3 marks**<br>• Stores data from all currently running programs concurrently in RAM<br>• Stops the data from overwriting each other in RAM/primary storage<br>• Decides which processes should be in main memory<br>• Makes efficient use of memory<br><br>*Process management:* **Max 3 marks**<br>• Allows one process to be paused whilst another process can be actioned<br>• Decides which process is to be run next<br>• Switches between processes to allow them to share the use of the processor<br>• Identification/description of scheduling | **4** |

## Answer 20

| | | |
|---|---|---|
| 8(a) | **1 mark** each to **max 2**:<br><br>• Creates an executable file<br>• … so the code can be tested multiple times without having to recompile<br>• … so **repeated testing** takes less time | **2** |

<table>
<tr>
<td>8(b)</td>
<td>1 mark for identification of each feature and 1 mark for matching description:<br><br>e.g.<br>For coding:
<table>
<tr><th>IDE feature</th><th>Description</th></tr>
<tr><td>Context-sensitive prompts</td><td>Gives suggestions for code as the user types instead of having to write/remember the code</td></tr>
<tr><td>Auto-correct</td><td>Corrects spelling mistakes so that user has fewer errors to correct</td></tr>
</table>
For presentation:
<table>
<tr><th>IDE feature</th><th>Description</th></tr>
<tr><td>Pretty-printing</td><td>Colour code keywords so the user can identify any errors</td></tr>
<tr><td>Expand/collapse (code) blocks</td><td>The user can hide code that they are not currently working on</td></tr>
</table>
For debugging:
<table>
<tr><th>IDE feature</th><th>Description</th></tr>
<tr><td>Single stepping</td><td>Run the code one line at a time // shows the effect of each line of code</td></tr>
<tr><td>Breakpoints</td><td>Stop the code running at a set point to check the flow/variable contents</td></tr>
</table>
</td>
<td>6</td>
</tr>
<tr>
<td>8(c)</td>
<td>1 mark each to max 2:<br><br>
• Saves programming/testing time as code does not have to be written/re-written from scratch // code does not have to be tested<br>
• Code is already tested so it is more robust/likely to work<br>
• The programmer does not need to maintain the library // library routines are updated automatically<br>
• Can perform complex calculations that the programmer may be unable to do<br>
• Makes code more easily readable</td>
<td>2</td>
</tr>
</table>

## Answer 21

| | | |
|---|---|---|
| 7(e)(i) | **1 mark** for:<br><br>Pre-written code/functions/routines that can be **imported/called** in another program | **1** |
| 7(e)(ii) | **1 mark** each to **max 4**:<br><br>• Maintenance not needed to be done by the programmer<br>• ... because the DLL is separate from program<br>• The calling program does not need recompilation by the programmer when a DLL file changes<br>• ... because the DLL file can be updated independently of the calling program<br>• … updates will apply to all programs that use the DLL file | **4** |

## Answer 22

| | | |
|---|---|---|
| 8(a) | **1 mark** per bullet point to **max 4**<br><br>• Reads/writes data to/from RAM<br>• … e.g. current data/instructions from a game so the CPU can access it<br><br>• Allocates virtual memory<br>• … when there is insufficient RAM to run a program/game<br><br>• Allocates RAM to optimise performance<br>• Paging<br>• Segmentation | **4** |
| 8(b) | **1 mark** per bullet point to **max 2**<br><br>• Software will have been built using a compiler // the software is pre-compiled<br>• Software is an executable file // the game is already in machine code // the game is already set-up to run on the console<br>• Source code is not provided so does not need compiling/interpreting | **2** |

## Answer 23

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>9(a)</td><td>1 mark for each correctly identified utility program<br><br>
<table>
<tr><th>Description</th><th>Utility program</th></tr>
<tr><td>Reorganises files on a disk to improve efficiency</td><td>Defragmentation software</td></tr>
<tr><td>Scans a hard disk to identify bad sectors</td><td>Disk contents analysis / repair software</td></tr>
<tr><td>Prepares a hard disk for first use</td><td>Disk formatter</td></tr>
</table>
</td><td>3</td></tr>
</table>

## Answer 24

<table>
<tr><td>2(c)(i)</td><td>1 mark per bullet point to max 3<br><br>
• Disk contents analysis checks for errors/problems with the disk<br>
• Disk repair attempts to fix the errors<br>
• The disk formatter prepares the disk for (initial) use (again).</td><td>3</td></tr>
<tr><td>2(c)(ii)</td><td>1 mark per bullet point to max 3<br><br>
• Installation of device driver software<br>
• Managing interrupts / signals from the device<br>
• Sending control signals to the device<br>
• Control of buffers<br>
• Management of queues</td><td>3</td></tr>
</table>

## Answer 25

| | | |
|---|---|---|
| 4(a)(i) | **1 mark** per reason to **max 3**<br><br>• DDL file is only loaded into memory when required<br>• ... so the executable file for the game is smaller<br>• Changes/improvements in the DLL file are independent of the main program<br>• ... the game program will not need to be recompiled<br>• ... the game program will get the benefit of the updates automatically<br>• The same DDL file can be used in several game programs (at the same time)<br><br>• (DLL) routines are pre-written saving the developers time<br>• (DLL) routines are pre-tested so should be reliable<br>• Developers can take advantage of other programmers' expertise | **3** |
| 4(a)(ii) | **1 mark** per reason to **max 2**<br><br>• Game will not work if DDL is corrupted<br>• An external change to the DDL could stop the game working or change the way it works<br>• The DDL file must be present at run-time otherwise there is an error | **2** |

## Answer 26

| | | |
|---|---|---|
| 3(a)(i) | **1 mark** for each advantage, **1 mark** for a valid expansion to **max 2 · 2**<br><br>□ Code is already tested<br>□ ...so it is more robust/likely to work<br><br>□ Saves programming time<br>□ ...code does not have to be written/re-written from scratch<br><br>□ The programmer can use e.g. mathematical functions<br>□ ...that s/he may not know how to code<br><br>□ If there is an improvement in the library routine<br>□ ...the program updates automatically | **4** |
| 3(a)(ii) | **1 mark** per bullet point to **max 2**<br><br>□ A collection of self-contained (shared library) programs<br>□ ...that are already compiled<br>□ Linked to the main program during execution<br>□ Library program code is separate from the .EXE file<br>□ Library file only loaded into memory when required at run time<br>□ A DLL file can be made available to several applications (at the same time)<br>□ If DLL routine is updated the program that uses it will run the update | **2** |

| | | |
|---|---|---|
| 3(b)(i) | **1 mark** per bullet point to **max 1**<br><br>▯ Errors can be corrected as they occur<br>▯ Can run a partially complete program when developing<br>▯ The effect of any change made to the code can be seen immediately | **1** |

## Answer 27

| | | |
|---|---|---|
| 1(a) | **1 mark** for each correct entry<br><br>**Management task** / **Description**<br>**Memory management** / Handles the allocation of memory to processes // Ensures two programs do not attempt to use the same memory locations // Keeps track of allocated and free memory locations<br>Security management / **Provides user accounts and passwords**<br>Interrupt processing / **Handles the signals sent when the attention of the processor is required elsewhere**<br>**Provision of a software platform** / Provides an environment within which programs can be run | **4** |

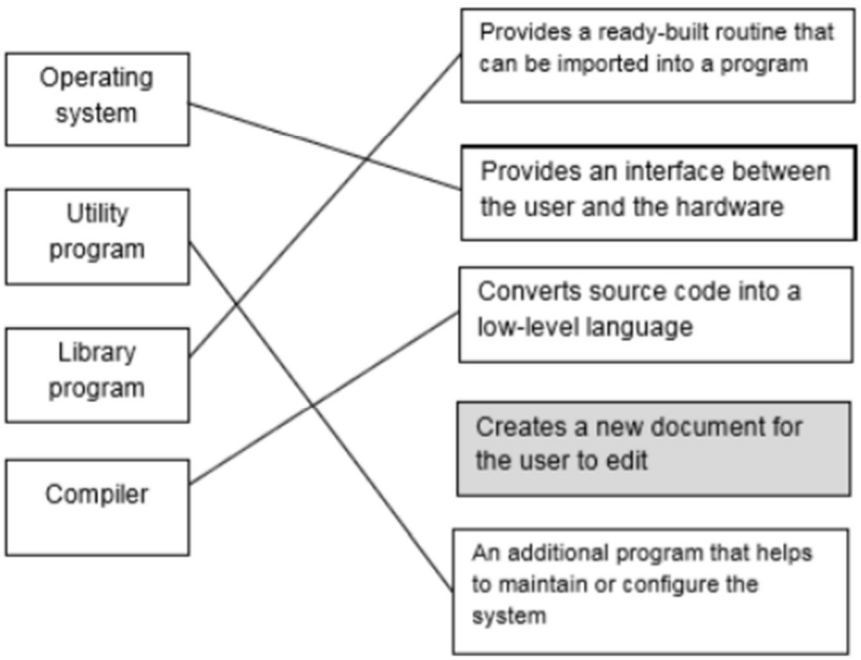
| | | |
|---|---|---|
| 1(b)(i) | **1 mark** per bullet point to **max 2** for formatter, **max 2** for defragmenter<br><br>**hard disk formatter**<br>▯ Makes existing data inaccessible<br>▯ Partitions the disk into logical drives<br>▯ Sets up the (specified) file system<br>▯ Prepares the disk for initial use<br>▯ May check for errors on the disk<br><br>**hard disk defragmenter**<br>▯ Re-organises the disk contents<br>▯ Moves split files so they are contiguous<br>▯ Creates a larger area of (contiguous) free space | **4** |
| 1(b)(ii) | **1 mark** per bullet point<br><br>For example:<br><br>▯ Backup software<br>▯ File compression<br>▯ Virus checker<br>▯ Disk contents analysis / repair | **3** |

## Answer 28

| | | |
|---|---|---|
| 2(b) | **1 mark** per bullet point to **max 4** for each management task, **max 6** in total<br><br>Process management:<br>• Manages the **scheduling** of processes<br>• … allows multi-tasking / multi-processing<br>• … ensures fair access<br>• … handles priorities<br>• Manages the resources the processes need<br>• Enables processes to share information<br>• Prevents interference between processes// resolution of conflicts<br><br>Provision of a user interface:<br>• Allows a user to **communicate** with the hardware // vice-versa<br>• … by making navigation around the system easier<br>• Provides facility for user inputting data<br>• Provides facility for outputting to the user<br>• By example e.g. command line / GUI / menu-driven | **6** |

| | | |
|---|---|---|
| 2(c) | **1 mark** per bullet point to **max 3** for each utility program, **max 4** in total<br><br>Virus checker:<br>• **Scans** files stored on a **computer system** for malicious code<br>• Scans files when they **enter the system** / memory stick inserted / download etc.<br>• Sets up a schedule for virus-checking<br>• Isolates / quarantines / deletes viruses<br>• Regularly updates the virus definitions | **4** |
| | Backup software:<br>• Creates a copy of the contents of a disk / partition. Can be set up to automatically backup // schedules backups<br>• Allows the user to decide what is backed up, e.g. all data // all files that have changed since the last backup<br>• Allows the user to set up an off-site backup<br>• May encrypt the backup files<br>• Restores the data if necessary | |
| 2(d) | **1 mark** per bullet point to **max 2**<br><br>• The code is translated one line at a time<br>• … and executed immediately<br>• The interpreter stops as soon as it finds an error | **2** |

## Answer 29

| | | |
|---|---|---|
| 1(a) | **1 mark** per correct line<br><br>**Type of software** / **Description**<br><br>  <br> | **4** |
| 1(b) | **1 mark** per bullet point to **max 3**<br><br>Disk repair:<br>▯ Checks for any errors / inconsistencies / bad sectors on the disk<br>▯ Resolves any errors on the disk<br>▯ Retrieves files / data from a damaged disk // re-constructs directory // recovers disc when data corrupt<br>▯ Marks bad sectors on the disk // marks bad sectors as unusable | **3** |

## Answer 30

| | | |
|---|---|---|
| 2(a) | **1 mark** per bullet point to **max 4** for each management task, **max 6** in total<br><br>**Process Management**<br>• Manages the **scheduling** of processes<br>• … allows multi-tasking / multi-processing<br>• … ensures fair access<br>• … handles priorities<br>• Manages which resources the processes require<br>• Enables processes to share information<br>• Prevents interference between processes // resolution of conflicts<br><br>**Memory Management**<br>• Allocates memory to processes<br>• Ensures fair usage of memory<br>• Organises memory / by example<br>• Makes use of virtual memory<br>• Keep processes separate<br>• To release memory when a process stops | 6 |

## Answer 31

| | | |
|---|---|---|
| 3(a)(i) | **1 mark** per bullet to **max 3**<br>• Allocates / deallocates RAM to programs/tasks/processes<br>• Keeps track of allocated and free memory locations<br>• Swaps data to and from the hard drive<br>• Handles virtual memory<br>• Paging // segmentation<br>• Memory protection, preventing a process accessing memory not allocated to it | 3 |
| 3(a)(ii) | **1 mark** per bullet to **max 3**<br>• Sets up user accounts<br>• Checks usernames, passwords // Authentication<br>• Implements access rights<br>• Automatic backup<br>• System restore / roll back (to previous stable state) | 3 |
| 3(a)(iii) | **1 mark** per bullet to **max 2**<br>• Device / peripheral management<br>• File management<br>• Process management<br>• Input / output management<br>• Error detection / recovery<br>• Provides a user interface<br>• Facilitates communication between hardware and software / hardware devices | 2 |

| | | |
|---|---|---|
| 3(b) | **1 mark** for each correct box ticked<br><br>**Programs** / **True** / **False**<br>Disk Defragmenter / ✓ /<br>Word Processor / / ✓<br>Library program / / ✓<br>Compression Software / ✓ / | **4** |

## Answer 32

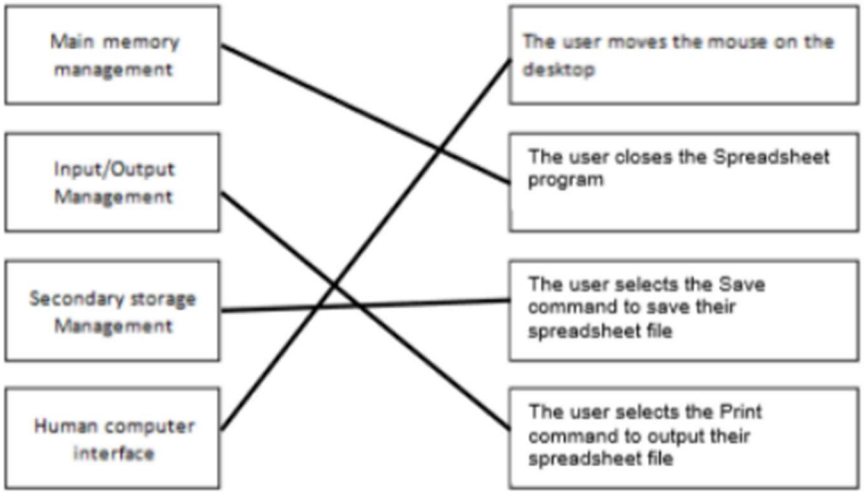
| | | |
|---|---|---|
| 1(a)(i) | **1 mark** per bullet to **max 3**<br>- Storage space divided into file allocation units<br>- Space allocated to particular files<br>- Maintains/creates directory structures<br>- Specifies the logical method of file storage (e.g. FAT or NTFS)<br>- Provides file naming conventions<br>- Controls access // implements access rights // implements password protection // Makes file sharing possible<br>- Specifies tasks that can be performed on a file (e.g. open, close, delete, copy, create, move etc.) | **3** |
| 1(a)(ii) | **1 mark** per bullet to **max 3**<br>- Installs printer driver<br>- Sends data to the printer / buffer to print // sends documents to the print queue<br>- Sends commands to printer<br>- Receives and handles (error) messages/signals/interrupts from the printer | **3** |
| 1(b)(i) | **1 mark** for each correct box ticked.<br><br>**Program** / **True** / **False**<br>Database / / ✓<br>Virus checker / ✓ /<br>Web browser / / ✓<br>Backup software / ✓ / | **4** |
| 1(b)(ii) | **1 mark** for each valid utility program to **max 2**<br>e.g.<br>- System clean up<br>- Automatic update<br>- Disk contents analysis / Disk checking / Disk repair<br>- File compression<br>- Disk formatter<br>- Firewall<br>- Disk Defragmenter | **2** |

## Answer 33

| | | |
|---|---|---|
| 6(b)(i) | **1 mark** per bullet to **max 2**<br><br>• Pre-existing / pre-compiled / pre-written modules / code ...<br>• ... can be linked into her program (without amendment)<br>• To perform common / complex tasks | **2** |
| 6(b)(ii) | **1 mark** per bullet point. **Max 2** for <u>one benefit</u>, **max 2** for <u>one drawback</u><br><br>Benefit:<br>• Less code needs to be written<br>• ... saves time / saves re-inventing the wheel<br><br>• Pre-tested // Used by many people<br>• ... reduces time testing // can be fairly sure that the function will perform as it should<br><br>• Can be written in a different programming language<br>• ... making use of special features of that language<br><br>• Can be complex algorithms (e.g. mathematical/graphics functions)<br>• ... she does not need to work out how to write it // ...that she may not know how to code<br><br>• Simplifies the program<br>• ... since just the name of the function included in the source code<br><br>Drawback:<br>• Compatibility issues<br>• ... may not work with the other code/may require changing program for it to work<br><br>• Not guaranteed thorough testing<br>• ... may be unknown or unexpected bugs / virus<br><br>• Library routine may not meet exact needs<br>• ... may give unexpected results // ... may need editing<br><br>• If library routine is changed<br>• ... there may be unexpected results / errors | **4** |

| | | |
|---|---|---|
| 6(c) | **1 mark** per bullet point. **Max 3** marks for interpreter, **max 3** marks for compiler<br><br>Interpreter:<br>• Used during development<br>• Debugging is easier<br>• … Because errors are reported as they are found // No need to wait until the end of the process for the error report<br>• … Because errors can be corrected as they are found<br><br>Compiler:<br>• Compiler used when development complete // compiler used when program ready for distribution<br>• Produces an executable file (.exe)<br>• … After compilation the compiler does not need to be present for the program to run<br>• … The program can be given to others without access to (source) code<br>• … Final program does not need to be re-compiled each time it is run<br>• Cross-compilation, the program can be compiled to run on different platforms | **4** |

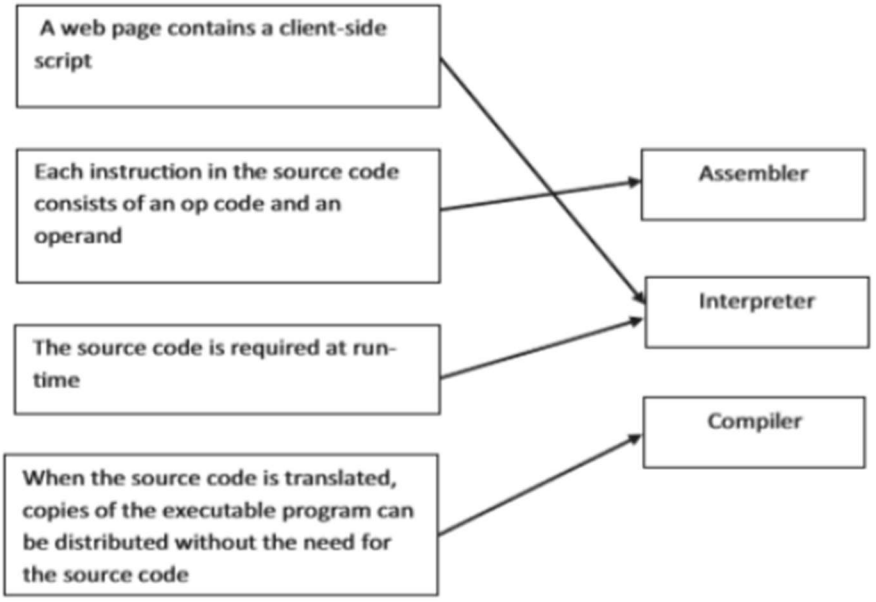
## Answer 34

| | | |
|---|---|---|
| 6(a) |  <br>**One** mark for each correct line from each left hand box **to max three** marks. | **3** |
| 6(b)(i) | File compression software | **1** |
| 6(b)(ii) | Backup software | **1** |
| 6(b)(iii) | Disk repair software | **1** |
| 6(b)(iv) | Anti-virus software | **1** |

## Answer 35

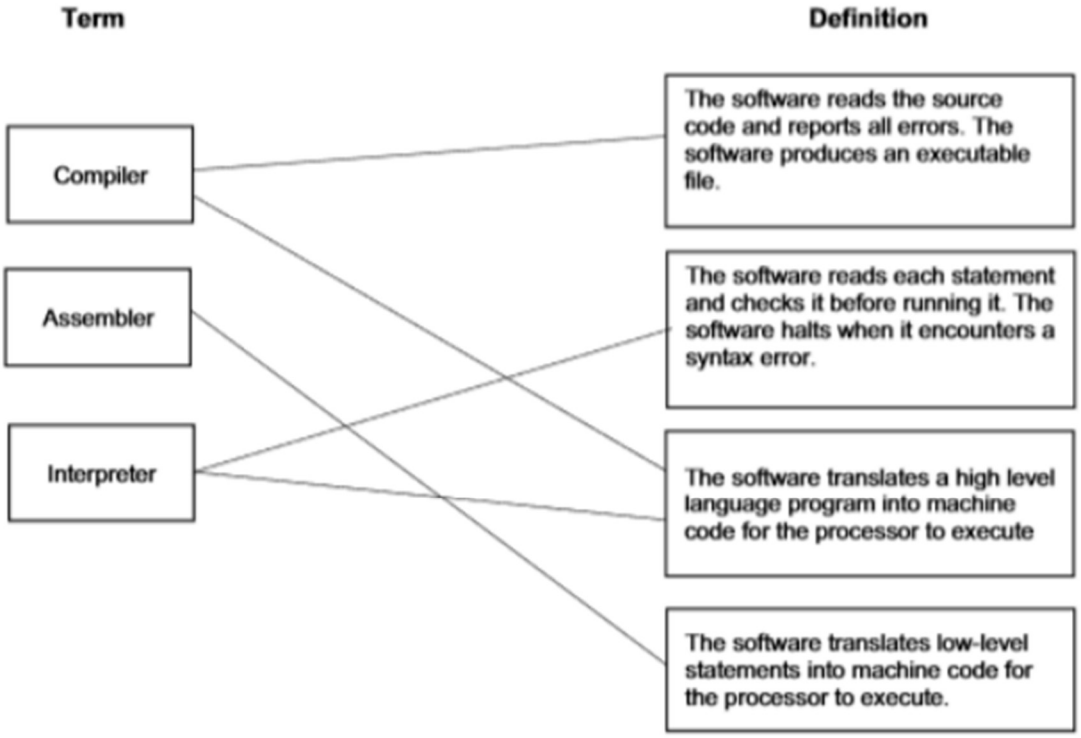
| | | |
|---|---|---|
| 4(a)(i) | **Two** from:<br>- The hardware is unusable without an OS // hides complexity of hardware from user 1<br>- Acts as an interface/ controls communications between user and hardware / hardware and software // or by example 1<br>- Provides software platform / environment on which other programs can be run 1 | **2** |
| 4(a)(ii) | **One** mark for the name and **one** mark for description.<br>**Max two** management tasks.<br><br>- Provides the Human Computer Interface (HCI) 1<br>Controls communications between user and hardware// or by example 1<br><br>- Main memory management 1<br>Memory protection to ensure that two programs do not try to use the same space // Use of virtual memory // Location of processes within the memory // By example 1<br><br>- File / Secondary storage management 1<br>Maintains directory structures // Provides file naming conventions // Controls access 1 | **Max 4** |
| | - Peripheral / hardware / device / Input-Output management 1<br>Installation of appropriate driver software // Controls access to data being sent to/from hardware/peripherals // Controls access to hardware/peripherals // manages communication between devices. 1<br><br>- Interrupt handling 1<br>Identifies priorities of interrupts // Saves data on power outage // Loads appropriate Interrupt Service Routine (ISR) // By example 1<br><br>- Security management 1<br>Makes provision for recovery when data is lost // Provides usernames and passwords // Prevents unauthorised access // Ensures privacy of data 1 | |
| 4(b)(i) | File compression software | **1** |
| 4(b)(ii) | Backup software | **1** |
| 4(b)(iii) | Disk defragmenting software | **1** |
| 4(b)(iv) | Anti-virus software | **1** |

## Answer 36

| | | |
|---|---|---|
| 2(a) |  | 4 |
| 2(b)(i) | **One mark** from:<br><br>▯ The program code can be translated to run on any processor / platform<br>▯ Source code is translated into machine independent intermediate code not machine dependent code | 1 |
| 2(b)(ii) | **Two marks** from:<br><br>▯ Java uses a two-step translation process<br>▯ Java code is partially interpreted – partially compiled<br>▯ Code is translated first into intermediate code / "bytecode"...<br>▯ ...using the Java compiler<br>▯ The bytecode is finally interpreted by the Java Virtual Machine | **Max 2** |

## Answer 37

1 **One mark** for each box on the left. **[3]**




*Marks allocated as follows:*

## Answer 38

1 **Four** from: **[4]**

- Compiler creates an executable//an interpreter does not create an executable.
- The compiled program can be independently distributed.
- Compiler reports all errors at the end of compilation//an interpreter stops when it reaches an error.
- Interpreter executes each statement immediately after decoding/checking it//a compiler checks the whole program for errors.
- The interpreter software/source code must be present in main memory every time the program is executed//the compiled program does not require compiler/source code to be present.
- Cross-compilation is possible/compile on one hardware platform to run on another.

## Answer 39

**(b) (i)** Any **two** from

- The hardware is unusable without an OS // hides complexity of hardware from user
- Acts as an interface / controls communications between user and hardware / hardware and software
- Provides software platform / environment on which other programs can be run [2]

**(ii)** Any **two** from:

- Process / task / resource management
- Main memory management
- Peripheral / hardware / device management
- File / secondary storage management
- Security management
- Provision of a software platform / environment on which other programs can be run – only if not given in part (b)(i)
- Interrupt handling
- Provision of a user interface run – only if not given in part (b)(i) [2]

**(c)** Any **two** from:

- A DLL file is a shared library file
- Code is saved separately from the main .EXE files
- Code is only loaded into main memory when required at run-time
- The DDL file can be made available to several applications (at the same time) [2]

## Answer 40

7 (a) **One mark** for the name and **one mark** for the explanation for **three** utility programs

- Disk formatter
- Prepares a hard disk to allow data to be stored on it

- Virus checker
- Checks for viruses and then quarantines removes any virus found

- File compression
- Reduces file size by removing redundant details (lossy / lossless)

- Backup software
- Makes copy of files on another medium in case of corruption / loss of data

- Firewall
- Prevents unauthorised access to computer system from external sources [6]

## Answer 41

7 **ONE** mark per bullet point, **MAX TWO** marks per task.

- Process / resource management
- Scheduling of processes / multi-tasking / multi-programming etc.
- Resolution of conflicts when two or more processes require the same resource

- Main memory management
- Memory protection to ensure that two programs do not try to use the same space
- Use of virtual memory
- Deciding which processes need to be in main memory at any one time
- Location of processes within the memory
- By example, e.g. when process terminates, memory is made available

- Peripheral / hardware / device management
- Installation of appropriate driver software
- Controls access to data being sent to / from hardware / peripherals
- Controls access to hardware / peripherals
- Manages communication between devices / hardware and software

- File / secondary storage management
- Maintains directory structures
- Provides file naming conventions
- Controls access

- Security management
- Makes provision for recovery when data is lost
- Provides usernames and passwords / encryption / user accounts
- Prevents unauthorised access
- Ensures privacy of data

- Provision of a software platform / environment
- On which other programs can be run

- Interrupt handling
- Identifies priorities of interrupts
- Save current memory / process values / saves data on power outage
- Loads appropriate Interrupt Service Routine (ISR)
- Any relevant example

## Answer 42

8 (a) **ONE** mark for each bullet point from **MAX TWO** groups.

- The code is already written
- (So the programmer is not starting over again) which saves time

- The code will have been used by many people
- So it should be already thoroughly tested // relatively error-free

- The programmer can use, e.g. mathematical / graphics functions, etc. (may not know how to code)
- Can be sure that the function will perform as it should // simplifies the program.

- The code should conform to industry standards
- And therefore contribute towards a more robust program

[4]

(b) (i) **ONE** mark for each benefit, and **ONE** mark for a further expansion.

- The executable file is smaller / the executable does not contain all the library routines ...
- ... DLL files are only loaded into memory when required.

- Changes / improvements / error correction to the DLL file code are done independently of the main program...
- ... So there is no need to recompile the main program
- ... All programs using it will benefit

- A single DLL file can be made available to several application programs...
- ... Saving space in memory / easing the pressure on memory

[4]

(ii) **ONE** mark for each bullet point from **MAX ONE** group.

- The executable code is not self-contained ...
- ... the DLL file(s) needed to be included at run time.

- Appropriate (linking) software must be available at run-time ...
- ... to link / include / import the DLL files.

- The DLL file must be present ...
- ... otherwise (unable to find X.dll) errors

- Unexpected changes to the DLL file / corrupted DLL file ...
- ... could mean the program stops working as expected

- Malicious changes to the DLL file ...
- ... could install a virus on the user's computer / related files could be corrupted

[2]

## Answer 43

| Activity | First pass or second pass |
|---|---|
| any symbolic address is replaced by an absolute address | 2 |
| any directives are acted upon | 1 |
| any symbolic address is added to the symbolic address table | 1 |
| data items are converted into their binary equivalent | 1 |
| forward references are resolved | 2 |

[5]

## Answer 44

| Statement | Interpreter | Compiler |
|---|---|---|
| This translator creates an executable file | | ✓ |
| When this translator encounters a syntax error, game execution will halt | ✓ | |
| The translator analyses and checks each line just before executing it | ✓ | |
| This translator will produce faster execution of the game program | | ✓ |
| Use of this translator makes it more difficult for the user to modify the code of the game supplied to the user | | ✓ |

1 mark for each correct row [5]

## Answer 45

**(a)** 1 mark per point

**CLI**

- user types in instructions to open/launch an application
- usually a number of instructions need to be typed in
- user is in direct communication with the computer system
- user has to type in the commands each time they want to open/launch the application

**GUI**

- user interacts with the system by using icons
- user doesn't need to know where application resides in the computer
- the application is opened/launched by clicking on an icon using a mouse (e.g.)
- windows is an example of GUI interface [2]

**(b)** **CLI**

- programmer/technician } need to access system and communicate
- systems engineer } at system level

**GUI**

- end user } does not need computer knowledge
  } just click on the icon to launch the application [2]

## Answer 46

**(d)** Any **two** points from:

- OS will allow one user **at a time** to use the computer
- each approved user is identified by a user id/password
- allows multi-tasking
- provides security for user files/profiles **[2]**