1.1 Systems architecture (1) - front

Description and purpose of the CPU

The registers in the von Neuman architecture

Description of MAR

Description of MDR

Description of PC (register)

The CPU components

Description of ALU (CPU component)

Description of C (CPU componen

INSPECTION COPY

COPYRIGHT PROTECTED

1.1 Systems ar

- MAR (memory address register)
- MDR (memory data register)
- Program counter
- Accumulator

INSPECTION COPY

- Memory data register
- Store the data or instruction that has just been fetched from the address in the MAR

INSPECTION COPY

- ALU (arithmetic logic unit)
- CU (control unit)
- Cache

- Control unit
- Sends control signals to give components commands
- Sends timing signals to coordinate actions

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

1.1 Systems architecture (2) - front

Description and affect of cache

Stages in FDE cycle

Fetch stage of FDE

Decode stage of FDE

Execute stage of FDE

Factors affectin CPU performanc

Description and impact of clock speed

Description and impact of the number of processor cores

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

1.1 Systems ar

- Fetch instructions/data
- Decode instructions/data
- Execute instructions

Decode instructions or data

Fe
m

- Clock speed
- Cache size
- Number of cores

Ex
in

- A core = one processor
- Each core runs one FDE cycle at a time. Two cores can run two cycles at a time.
- The more cores, the more FDE cycles can be run each second.

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

1.1 Systems architecture (3) - front

Features of an embedded system

Benefits of an embedded syste

Drawbacks of an embedded system

Description and examples of registers

Difference between data and an address in storage

Examples of embedded systems

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

1.1 Systems ar

- No excess hardware (only what is needed is included). This reduces cost
- Fewer components means less going wrong
- Faster to load (less bloatware, fewer excess features, etc.)

- Extremely fast but small piece of memory
- Directly accessible by the CPU
- Examples: MAR, MDR, PC, ACC

C
ve
ar

- Washing machine
- Components in a car (e.g. lane diversion warning)
- Digital watch

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

1.2.1 Memory - front

Description of primary memory

Examples of primary memory

Description of RAM

Description of ROM

RAM vs ROM

Need for primary memory

Description of virtual memory

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

INSPECTION COPY

- **RAM**
- **ROM**
- **Cache**
- **Virtual memory**

INSPECTION COPY

- **Read-only memory**
- **Memory that stores data that cannot be altered, such as boot up instructions. In an embedded system it can store all program instructions.**
- **ROM is non-volatile (data *is not* lost when device has no power)**

INSPECTION COPY

- **Memory to be accessed by the CPU**
- **To store data to load the computer**
- **To store data temporarily for the processor to accesss**

INSPECTION COPY

COPYRIGHT PROTECTED

1.2.2 Storage (1) - front

Purpose of and need for secondary storage

Factors to consider when choosing a storage device

Types of storage media

Description of optical storage

Examples of optical storage

Description of magnetic storag

Examples of magnetic storage

Description of solid-state storage

INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
COPYRIGHT PROTECTED
Zig Zag Education

1.2

- Capacity
- Speed
- Portability
- Durability
- Reliability
- Cost

INSPECTION COPY

- Marks are made on the surface of a disk
- Light is reflected on the marks using a laser
- The reflection is read as 1 or 0

INSPECTION COPY

- The surface of a disk is magnetised
- The electromagnetic current is read as 1s and 0s

INSPECTION COPY

- Device is made of logic gates (NAND gates)
- Data is held electronically (i.e. not using moving parts)

INSPECTION COPY

COPYRIGHT PROTECTED

1.2.2 Storage (2) - front

Examples of solid-state storage

Characteristics o optical storage

Characteristics of optical storage

Characteristics o solid-state storage

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

1.2

- Low capacity
- Low access speed
- Small in size so portable
- Not very durable (easily damaged)
- Fairly reliable
- Cheap per GB

- Fairly large capacity, ever increasing
- Very fast access speed
- Very portable
- Very durable
- Reliable, although with limited read / write times
- Expensive per GB

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

1.2.3 Units (1) - front

Calculation used to estimate file size of an image

Calculation used to estimate the file size of a sound file

Unit which consists of 1,000 bytes

Why data is stored in binary

Calculation used to estimate file size of a text file

Description of metadata and us in file size calculations

Unit which consists of 1,000 kilobytes

Unit which consists of 1,000 megabyte

INSPECTION COPY

COPYRIGHT PROTECTED

1

Samples per second ×
Bits per sample (bit rate) ×
Duration

N
N
N

- Computers are made of logic gates/switches
- They can only be in 2 states (on and off)
- These relate to binary 1 and 0

Ki

- Data about the data in the file; e.g. model of camera used to take a photograph
- File size calculations can have 10% added to allow for this metadata

  (This is not required in your calculations)

N
N

Gigabyte

M

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

1.2.3 Units (2) - front

Unit which consists of 1,000 gigabytes

Number of bits in a nibble

Unit which consists of 1,000 terabytes

Number bits in a byte

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

1

4

Te

8

Pe

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

1.2.4 Data storage (1) - front

Description of a bit

Description of a nibble

Description of a byte

Description of a kilobyte (KB)

Description of a megabyte (MB)

Description of a terabyte (TB)

Description of a petabyte (PB)

Convert 29 into binary

INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
COPYRIGHT PROTECTED
Zig Zag Education

1.2.4 Da

- 4 bits
- e.g. 1000, 1111, 0101

1

1000 bytes

8

e.

1000 GB

10

1 01

10

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

1.2.4 Data storage (2) - front

Convert 136 into binary

Convert 254 into binary

Smallest number represented by 8 bits

Largest number represented by 8 bits

How many numbers can be represented by 8 bits?

Convert 599 into binary

Convert 756 into binary

Convert `1100101` into denary

INSPECTION COPY

COPYRIGHT PROTECTED

1.2.4 Da

11111110

10

255

0

1001010111

25
nu

2[illegible]

10

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT
PROTECTED

1.2.4 Data storage (3) - front

Convert 01101 into denary

Convert 100110 into denary

What is binary 0 + 0?

What is binary 1 + 0?

What is binary 0 + 1?

What is binary 1 + 1?

What is binary 1 + 1 + 1?

Add the followin binary numbers:

00011010
01101011

INSPECTION COPY

COPYRIGHT PROTECTED

INSPECTION COPY

38

13

1

0

INSPECTION COPY

0 carry 1

1

```
00011010
01101011
10000101
1111 1
```

1

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

1.2.4 Data storage (4) - front

Add the following binary numbers:

001101
10001[illegible]

Error that occurs whe these binary numbers are added together:

11000101
1111111

Description of binary addition overflow

Description of a right binary shif

Description of a left binary shift

Perform a 1-space left shif on 100101

Perform a 2-space left shift on:

01101[illegible]1

Perform a 4-spa left shift on:

10110101

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

1.2.4 Da

0
1
1
1

Overflow – the result is too large to be represented in 8 bits

- **Move the bits of a binary number a set number of spaces to the right**
- **Fill in the new gaps on the right with 0s**
- **Each shift left multiplies the number by 2**

Th is re nu

1001010

1001101010000

01

INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
COPYRIGHT PROTECTED
Zig Zag Education

1.2.4 Data storage (5) - front

Perform a 1-space right shift on:

100101

Perform a 2-spa right shift on:

01101011

Perform a 4-space right shift on:

100110101

Description of hexadecimal

Hexadecimal value of 10

Hexadecimal value of 11

Hexadecimal value of 12

Hexadecimal value of 13

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

1.2.4 Da

00011010

01

- Base-16 data representation
- Numbers for 0–9; 10–15 are represented by letters (A–F)

00

B

A

D

C

INSPECTION COPY

COPYRIGHT PROTECTED

1.2.4 Data storage (6) - front

Hexadecimal value of 14

Hexadecimal value of 15

Steps to convert a binary number to hexadecimal

Steps to convert denary number to hexadecimal

Convert `10101110` into hexadecimal

Convert `01010000` into hexadecimal

Convert `1110010` into hexadecimal

Convert 156 into hexadecimal

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

1.2.4 Da

F

- Divide by 16 for the first digit; the remainder is the second digit
- Replace 10 with A, 11 with B, etc.

```
0101   0000
  5      0
```

= 50

156 / 16 = 9 remainder 12
= 9 remainder C
= 9C

E

1

1

INSPECTION COPY

COPYRIGHT PROTECTED

1.2.4 Data storage (7) - front

Convert 58 into hexadecimal

Convert 218 into hexadecimal

Definition of a character set

Examples of character sets

Description of ASCII

Description of UNICODE

Benefit of ASCII over UNICODE

Benefit of UNICODE over ASCII

INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
COPYRIGHT PROTECTED

1.2.4 Da

218 / 16 = 13 remainder 0

[illegible] remainder A

= DA

- ASCII
- Unicode

Th
be
co

- Character set
- Allows for 17 x 65,536 characters (over 1.1 million in total)
- Has become the universal standard, as it supports languages used worldwide.

More bits per character = more characters can be represented

Fe
=
le
=

INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
COPYRIGHT PROTECTED
Zig Zag Education

1.2.4 Data storage (8) - front

Drawback of ASCII over Unicode

Drawback of Unicode over ASCII

If the ASCII value for A = 65, what is the ASCII value for D?

Why does 'A' not equal 'a'?

Description and features of a bitmap image

Description of a pixel

Description and impact of colour depth in an image

Definition of metadata

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

1.2.4 Da

More bits per character
= larger file size

INSPECTION COPY

- The character's numbers are compared
- Their characters have different character set numbers

INSPECTION COPY

A small square of a single colour

INSPECTION COPY

Data about data

Fe
=
re

68

INSPECTION COPY

COPYRIGHT
PROTECTED

1.2.4 Data storage (9) - front

Examples of image metadata

Definition of image resolution

Effect of increasing the colour depth

Effect of increasing the resolution of an image

Description of sound sampling

Description of sampling rate

Impact of increasing sampling rate

Impact of decreasing sampling rate

INSPECTION COPY

COPYRIGHT
PROTECTED

1.2.4 Da

The number of pixels per square inch of the image

Da
e.
ar
co

INSPECTION COPY

- More pixels per inch
- = more data to store
- = larger file size

INSPECTION COPY

- The number of samples taken per second
- The larger the sampling rate the more accurate the sound file

INSPECTION COPY

- Less accurate recording – sound is less like the original
- Smaller file size means fewer samples, so less binary being recorded

INSPECTION COPY

COPYRIGHT PROTECTED

1.2.4 Data storage (10) - front

Description and impact of sound sampling bit depth

Impact of increasing soun

sampling bit dep

Description and impact of sound duration

Impact of decreasing sour

sampling bit dep

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

1.2.4 Data

- + A greater range of sounds can be recorded and stored…
- − More bits per sample, so larger file size

- A smaller range of sounds can be recorded and stored…
- but fewer bits per sample = smaller file size

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT
PROTECTED

Zig Zag Education

1.2.4 Data storage (11) - front

Purpose of compression

Description of lossy compression

Description of lossless compression

Benefit of lossy compression ove losless compression

Benefit of lossless compression over lossy compression

Drawback of loss compression ove lossless compression

Drawback of lossless compression over lossy compression

Examples of whe lossy and lossles compression is used

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

1.2.4 Data

- The decompressed file is not the same as the original
- Some data is lost
- Used where the data loss is not noticable, e.g. images, sound, video

Te
fil

-
-
-
-
-

Achieves greater compression – the file sizes can be made smaller than in lossless compression

Data loss can be noticed, e.g. a text document, or computer program cannot afford to lose content

No
ori
re

- Lossy: images, sound, video
- Lossless: text files, program code

Th
re
ex

INSPECTION COPY

COPYRIGHT PROTECTED

1.3.1 Networks and topologies (1) - front

Features of a LAN

Features of a WA

LAN vs WAN

Factors that affe network performance

Description of a client–server network

Benefits of a client–server network

Drawbacks of a client–server network

Benefits of a pe to-peer network

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

1.3.1 Networks and t

- Wide area network
- Large geographical area
- Uses public infrastructure (e.g. phone lines, satellites)

- Bandwidth (how much data can be transmitted at a time)
- Number of devices connected
- Amount of traffic (how much data is being transferred across it)
- Hardware being used (including type of cable)

- Central file storage (and backup)
- Users can log onto any client and access same data
- Central security management
- Central control
- Central monitoring

- Can store documents, software, etc.
- Computers are independent and can be removed from the network and still used

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

1.3.1 Networks and topologies (2) - front

Drawbacks of a peer-to-peer network

Examples of network hardwa

devices

Purpose of a wireless access point

Purpose and function of a router

Purpose and function of a switch

Purpose and function of a NI

Purpose and function of WNIC

Purpose and function of a modem

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

1.3.1 Networks and

- Router + Modem
  (often combined)
- Switch / Hub
- NIC (network interface controller) / WNIC (wireless NIC)
- Copper / Ethernet cables
- Fibre-optic cables
- Wireless access point

- Hardware which connects devices in a network (using cables or wirelessly)
- Receives signals from devices and forwards them to their destination
- Remembers addresses of devices attached to it
- Often has integrated modem for internet access

- Network interface card/controller
- Hardware device
- Within a computer (can be added to a computer)
- Allows Ethernet cables be connected for access to a network

- Modulator-demodulator
- Allows connection to the Internet
- Converts analogue phone line data to digital computer data and back again

INSPECTION COPY

COPYRIGHT PROTECTED

1.3.1 Networks and topologies (3) - front

Purpose and function of hub

Description of the Internet

Purpose of IP

Description of WWW

Purpose of URL

Function of DNS

Description of 'the cloud'

Device that host a website

INSPECTION COPY

COPYRIGHT PROTECTED

1.3.1 Networks and t

- Worldwide network made up of smaller networks (e.g. WAN, LAN)
- The infrastructure that connects computers, e.g. telephone lines

- World Wide Web
- The websites and resources hosted on the Internet

- Domain name server
- Browser sends URL to DNS
- DNS has database of domains and matching server IPs
- DNS finds domain in database *
- DNS returns server IP to browser

* If DNS does not find IP, it forwards request to higher DNS. Higher DNS then returns IP to original DNS

Web server

Sc
da
on

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

1.3.1 Networks and topologies (4) - front

Drawbacks of cloud storage	**Benefits of clou storage**
Description of web client	**Description of web server**
Description of network topology	**Features of star topology**
Benefits of star topology	**Drawbacks of star topology**

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

1.3.1 Networks and t

- Files/data can be accessed anywhere with Internet access
- Easily increase storage size
- Backup and security managed externally

Computer connected to the Internet that stores web pages, files and data that can be accessed remotely

A
re
ov
di
da

- There is a central computer
- All computers are directly connected to the central computer
- All data transferred is sent into central computer and then out to its destination

- If the central computer goes down, the entire network goes down
- Limited number of computers before the network slows significantly
- Installation can be costly due to cabling required

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

1.3.1 Networks and topologies (5) - front

Features of mesh topology

Benefits of mesh topology

Drawbacks of mesh topology

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

1.5 Networks and t

- Can manage a high level of network traffic
- If one computer goes down, the others don't
- New devices can easily be added

-
-
-

Se
is
tii

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

1.3.2 Wired and wireless networks, protocols a

Purpose and function of Ethernet cable

Purpose and function of fibre-optic cable

Description and impact of Wi-Fi frequency

Definition of a Wi-Fi channel

Benefits of fibre-optic

Definition of encryption

Drawbacks of fibre-optic

Methods for aiding Wi-Fi security

INSPECTION COPY

INSPECTION COPY
COPYRIGHT PROTECTED
Zig Zag Education

1.3.2 Wired and wireless networks, protocols

- Cable that transmits light signals
- Faster transmission than copper cable

A set of frequencies used in the transmission of data

Jumbling up of data so if it is read without knowing the key to decrypting it, the data will not make sense

- Anyone can intercept Wi-Fi signals if unsecured
- Encrypt data when sending
- Use password on router to allow decryption of messages
- Limit access to specific MAC addresses

INSPECTION COPY

COPYRIGHT PROTECTED

1.3.2 Wired and wireless networks, protocols a

Purpose of encryption

Description of IPv4

Description of IPv6

Description of protocol

Description of Ethernet

Purpose of an IP address

Description and purpose of MAC address

Use of TCP/IP

INSPECTION COPY

COPYRIGHT PROTECTED

1.3.2 Wired and wireless networks, protocols

- 32-bit address
- 4 sets of numbers
- Each set is 8-bits (0–255)
- Each set is separated by full stops (.)
- Usually represented in denary
- e.g. `192.168.0.1`

To
be
de
so
it
wi

- A set of rules, e.g. how to transfer data between 2 computers
- Used when transmitting data so that the sender and receiver are communicating in the same way

- 12
- Ea
- Ea
- U
- e.
  1
  a

Internet Protocol address

- A unique address assigned to a device connected to the Internet
- Used to identify the location of clients and servers, so that data can be transferred between them

- Transmission Control Protocol / Internet Protocol
- Allows transmission of data across the Internet
- Governs allocation and use of IPs

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

1.3.2 Wired and wireless networks, protocols a

Use of HTTP & HTTPs

Use of FTP
(protocol)

Use of IMAP
(protocol)

Use of POP
(protocol)

IMAP vs POP

Use of SMTP
(protocol)

Description of protocol layers

Benefit of protoc layers

INSPECTION COPY

INSPECTION COPY
COPYRIGHT PROTECTED
Zig Zag Education

1.3.2 Wired and wireless networks, protocols a

- File Transfer Protocol
- Governs the transfer of files online

- Post Office Protocol
- Allows a computer to retrieve emails from a server
- Stores the email on your computer and removes it from the server

- Simple Mail Transfer Protocol
- Allows emails to be sent

So one layer can be removed, edited, replaced, etc. without any other layers being affected

A
ap
tra
di
di
la
da
ph

INSPECTION COPY

COPYRIGHT PROTECTED

1.4 Network security (1) - front

Importance of keeping data secure

Threats to a network

Defintion and examples of malware

Definition of a virus

Definition of spyware

Description of social engineeri

Description of phishing

Description of pharming

INSPECTION COPY

COPYRIGHT PROTECTED

1.4 Networ

- Virus/malware
- Spyware
- Hacker
- People, e.g. shoulder surfing
- Brute force attack
- Data interception
- Poor network policy
- SQL injection

Da
pr
pr
st
m

- A piece of software
- … that can replicate itself
- … that can cause damage to a computer system or the data

- People's carelessness can cause problems, e.g. using weak passwords
- People can perform malicious acts e.g. shoulder surfing
- People can respond to phishing emails giving away personal data

Fraudulent websites that look like real ones. People are unknowingly directed to these pages and enter personal data such as passwords.

Fr
pe
cl
to
of
su

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT
PROTECTED

Zig Zag Education

1.4 Network security (2) - front

Definition of a brute force attack

Definition of data interception

Definition of a denial of service attack

Definition of an SQL injection

Description of penetration testing

Description and examples of physical security

Purpose of anti-malware software

Purpose of antivirus softwa

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

1.4 Netwo

Data is being transmitted somewhere, and someone other than the intended recipient accesses the transmission and reads the data

Sc
m
ac
sy
pa

- Entering SQL statements into a text box on a website
- When the form is submitted, the query is executed by the database and returns data meeting the query

-
-

- Methods of preventing an unauthorised person gaining physical access to a computer or network
- Examples: locks, CCTV, security guards, biometrics

Tr
ur
a
or
pc
fi

- When run, it scans a computer to find viruses and either removes them or quarantines them
- Scans downloaded files for viruses; warning the user of any present

W
co
ar
qu

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

1.4 Network security (3) - front

Purpose of anti-spyware software

Purpose of a firewall

Definition of a hacker

Description of user access leve

Features of a strong password

Description of encryption

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

1.4 Netwo

- A piece of software (or hardware) that monitors incoming and outgoing signals
- Can prevent unwanted incoming/outgoing signals
- Can block transmissions from unwanted/unknown IP address
- Can prevent hacking

A
ga
to

Giving people different permissions, e.g. some people cannot read files, some people can read but not change files and some people have full access

- Jumbling up of data so if it is read without knowing the key to decrypt it, the data will not make sense
- Prevents people understanding data if they intercept it

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT
PROTECTED

Zig Zag Education

1.5 Systems software (1) - front

Purpose and examples of systems software

Tasks performe by an operating system (OS)

Description of user interface

Types of user interface

Features of a GUI

Functions of memory management (O

Functions of multitasking (by an OS)

Functions of use management (O

INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
COPYRIGHT PROTECTED
Zig Zag Education

1.5 Systems

- User interface
- Memory management / multitasking
- Peripheral management / drivers
- User management
- File management

- GUI (graphical user interface)
- Command line

So
us
pr
to

- Deciding when to read/write data from/to memory and where to read/write it from/to
- Deciding when to use virtual memory, and managing VM
- Managing access to memory, e.g. RAM, hard drive
- Supports multi-tasking

- Setting up and managing user accounts, e.g. usernames and passwords
- Allocating files for users
- Access rights (read write, read-only, no access)

- O
o
- A
s
s
m
- It
th
d

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

1.5 Systems software (2) - front

File management functions (OS)

Purpose and examples of util software

Description of a fragmented disk

Process of defragmentatio

Purpose of data compression

Situations where data compressio is needed

Description and purpose of encryption

Purpose of file backup

INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
COPYRIGHT PROTECTED
Zig Zag Education

1.5 Systems

- Performs maintenance
- e.g. encryption, defragmentation, compression, backup

- Moves all data blocks together, and all free space together
- Increases efficiency when writing to and reading from the disk

- F fi s
- L p r
- D

- Downloading/uploading files – the smaller the file, the the faster the files can be transfered
- Streaming – the smaller the file, the less bandwidth is required by both the the server and client
- Limited storage size – the smaller the file, the more files can be stored on the device

To si

Creating a copy of files, in case the originals are lost

Ju if kn de no

INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
COPYRIGHT PROTECTED

1.7 Systems software (3) - front

Importance of backing up files

Types of backup

Description of full backup

Description of incremental backup

Advantage of full backup

Advantage of incremental backup

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

1.7 Systems

- Full
- Incremental

-
-

Only files, data and/or software that have changed since the last backup are backed up

Al
ar

Faster to run because once everything is backed up, only small parts are backed up each time

No
ch
fil
ba

INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
COPYRIGHT PROTECTED

1.6 Ethical, legal, cultural and environmental c

Definition of ethics

Laws relevant to Computer Scien

Definition of cultural issues

Definition of environmental issues

Definition of privacy issues

Features of proprietary software

Benefits of proprietary software

Drawbacks of proprietary software

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

1.6 Ethical, legal, cultural and environmental

- Data Protection Act / GDPR
- Computer Misuse Act
- Copyright Designs and Patents Act
- Creative Commons licensing
- Freedom of Information Act

Th
ac
rig

Th
de
of

INSPECTION COPY

Factors that impact on the environment, e.g. waste, use of electricity

Cu
en
of
sp

INSPECTION COPY

- Commercial software usually purchased from a company
- Source code is not available
- Restrictions apply, e.g. permitted number of users
- Cannot edit the program code; may come with many features you don't want/need
- Usually thoroughly tested and lots of support available

INSPECTION COPY

- Contains features you don't want/need
- You cannot edit to make it meet your needs

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

1.6 Ethical, legal, cultural and environmental c

Features of open source software

Benefits of open source software

Drawbacks of open source software

Features of the Data Protection Act 2018

Features of the Computer Misuse Act 1990

Features of the Copyright Desig and Patents Act 1988

Description of software licences

Types of software licence

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

1.6 Ethical, legal, cultural and environmenta

Can edit the program to make it do what you want/need

Protects personal data when held by organisations: Data must be collected, stored and processed lawfully, fairly and transparently; Data must be collected, stored and processed for the purpose it was originally collected; Only the data required must be collected, stored and processed; All data must be accurate; Data must be stored for a limited time for its intended use; Data must be collected, stored and processed securely.

Le
ac

Illegal to copy (copyrighted) software, run copied software or transmit software by electronic means

- Open source
- Proprietary

So
or

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

2.1 Algorithms (1) - front

Definition of computational thinking

Three elements of computationa thinking

Definition of abstraction

Benefits of abstraction

Examples of abstraction

Benefits of decomposition

Definition of an algorithm

Name two searching algorithms

INSPECTION COPY

COPYRIGHT PROTECTED

- Abstraction
- Decomposition
- Algorithmic thinking

- Clear focus on core tasks = better outcome
- Makes programs more user-friendly and secure
- Can increase efficiency of code; making the program require less memory and faster to run

- Reduce the problem's complexity
- Identify subroutines to program independently; can even be programming by different people and combined later to make a single program
- Identify where elements are repeated (saves time as only have to program once)

- Binary search
- Linear search

2.1 A

-
-

Re
un
in
a

-
-

Th
to

INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
COPYRIGHT PROTECTED
Zig Zag Education

2.1 Algorithms (2) - front

Use of searching algorithms

Linear search step by step

Binary search step by step

Perform a linear searc
for 20 in the data:

1, 55, 60, 88, 20, 100, 15,

Perform a linear search for 100 in the data:

1, 56, 99, 20, 22, 65, 3, 4

Why can't you do a binary search to find
in the data:

1, 55, 60, 88, 20, 100, 15,

Perform a binary search for 36 in the data:

1, 12, 18, [illegible]3, 36, 47, 50

Perform a binary sea
for 22 in the data:

1, 17, 22, 36, 45, 55, 67,

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

2.1 A

Items can be in any order

(1) Start at the first item

(2) Is it the item you want?
☑ go to (5); ☒ go to (3)

(3) Are there any more items?
☑ go to (4); ☒ go to (5)

(4) Move to next item. Go to (2)

(5) Stop

1×	55	60	88	20	100	15	2
1	55×	60	88	20	100	15	2
1	55	60×	88	20	100	15	2
1	55	60	88×	20	100	15	2
1	55	60	88	20✓	100	15	2

The data is not in order

Low			Mid				High	
1	17	22	35	45	55	67	50	22 <36 *take 1st half*
Low	Mid		High					
1	17	22	36	45	55	67	50	22 >17 *take 2nd half*
		Low	High					
1	17	22	36	45	55	67	50	22=22 *Item found*

To
lis

Ite
(1)
(2)
(3)
(4)
(5)
(6)

1×
1
1
1
1
1
1
1

No i

Lo

INSPECTION COPY

COPYRIGHT PROTECTED

2.1 Algorithms (3) - front

Three sorting algorithms

Use of sorting algorithms

Bubble sort

Perform a bubble sort the data:

1, 23, 6, 18, 44

Perform a bubble sort on the data:

22, 18, 37, 44, 55, 41

Merge sort step by step

Perform a merge sort on the data:

1, 23, 6, [illegible] 44, 7

Perform a merge sort the data:

22, 18, 33, 44, 41, 55, 4

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

2.1 A

Put data items in an array/list in ascending or descending order

●

●

●

1	23	6	18	44	7	*Unsorted*
1	6	23	18	44	7	*6>23, swap*
1	6	18	23	44	7	*23>18, swap*
1	6	18	23	7	44	*44>7, swap*
1	6	18	7	23	44	*23>7, swap*
1	6	7	18	23	44	*18>7, swap*
1	6	7	18	23	44	*Sorted*

(1)

(2)

(3)

(4)

(5)

- **Split all items into their own lists**
- **Merge each pair of lists, putting the data items in order**
- **Merge the new lists in pairs, putting the data items in order in the new lists**
- **Continue until there is a single list**

22	18	33	44	41	55	49

22 | 18 | 33 | 44 | 41 | 55 | 49

18 22 | 33 44 | 41 55 | 49

18 22 33 44 | 41 49 55

18	22	33	41	44	49	55

INSPECTION COPY

COPYRIGHT PROTECTED

2.1 Algorithms (4) - front

Insertion sort step by step

Perform an insertion sort on the data:

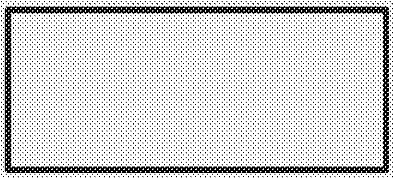
1, 23, 6, 18, 44, 7

Perform an insertion sort on the data:

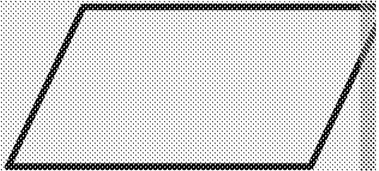
22, 18, 33, 44, 41, 55, 49

Features of a flow chart

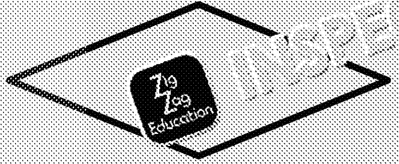
Function of this flow chart symbol:



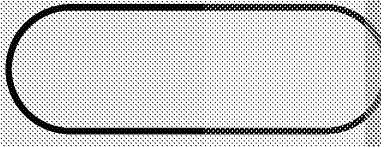
Function of this flow ch symbol:



Function of this flow chart symbol:



Function of this flow ch symbol:



INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

2.1 A

1	23	6	18	44	7
1	23	6	18	44	7
1	6	23	18	44	7
1	6	18	23	44	7
1	6	18	23	44	7
1	6	7	18	23	44

Ta
the
so
Ins
☒ l
lis

Ta
to
its
rig
co
ite

- Diagram that shows how the inputs, processes and outputs within a system
- Allows any programmer to understand and go on to develop the program

22
18
18
18
18
18
18

Flow chart input/output

Fl

- Flow chart terminator (contains the word START/STOP)
- 'START' is at the beginning of a flow chart
- 'STOP' is at the end of a flow chart

INSPECTION COPY

COPYRIGHT PROTECTED

2.1 Algorithms (5) - front

Exam reference language binary search function; return position if found and -1 if not found

Exam reference language linear search function; return position if found and -1 if n found

Exam reference language bubble sort procedure on global array

Efficient exam reference language bubble sort procedure o global array

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

2.1 A

```
function linearSearch(value, array)
  arraySize = array.length()
  x = 0
  valueFound = False
  while(x < arraySize OR
  valueFound == False)
    if(array[x] == value) then
      returnValue = x
      valueFound = True
    endif
    x = x + 1
  endwhile
endfunction
```

```
func
  ar
  lo
  up
  va
  wh

  en
  re
end
```

```
procedure bubbleSort()
  arraySize = array.length()
  x = 0
  swap = False
  while(swap == False and x < arraySize
   - 1)
    swap = False
    for y = 0 to arraySize - 1
      if(array[x] > array[x + 1]) then
        temp = array[x]
        array[x] = array[x + 1]
        array[x + 1] = temp
        swap = True
      endif
    next x
  endwhile
endprocedure
```

```
proc
  ar
  fo

  ne
endp
```

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

2.1 Algorithms (6) - front

Exam reference language insertion sort procedure on global array

Description of trace table

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

2.1 A

- Used to dry-run an algorithm
- Follow each step of the algorithm and fill in the variable values in a new row in the table as they change
- Check the values at each stage to work out where and when an error occurs

```
proc
ar
fo



ne
endp
```

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

2.2 Programming fundamentals (1) - front

Definition of a variable

Definition of a constant

Definition and examples of an operator

Purpose of input

Purpose of output

Three basic programming constructs

Description of a sequence

Description of selection

INSPECTION COPY

COPYRIGHT PROTECTED

2.2 Programming fun

- Space in memory
- Each one has a unique identifier
- Stores a value
- The value cannot change while the program is running

To read a value in from the user

A
ac

- Sequence
- Selection
- Iteration

To
sc

Code may or may not be run depending on a condition

A
ex
th

INSPECTION COPY

COPYRIGHT PROTECTED

2.2 Programming fundamentals (2) - front

Two types of selection

Example IF statement

Example SWITCH statement

Description of iteration

Two types of loop

Description of a count-controlled loop

Example FOR loop

Description of a condition-controlled loop

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

2.2 Programming fun

```
if condition is true then
   run this code
elseif this condition is
true then
   run this code
else
   run this code
endif
```

-
-

Repeating code until/while a condition is true or for a set number of times

```
sw
ca

ca

de

en
```

The loop states how many times it will run

-
-

The loop runs either while a condition is true or until a condition is true

```
FO
st

NE
```

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

2.2 Programming fundamentals (3) - front

Example
WHILE loop

Example
DO UNTIL loop

Function of the code
`string.length`

Function of the code
`LEFT(string,2)`

Function of the code
`RIGHT(string,2)`

Function of the code
`string.upper`

Function of the code
`string.lower`

Function of the code
`ASC(character)`

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

2.2 Programming func

```
# Condition-controlled loop;
# runs until the condition
# is true.  The code in the
# loop will always run once.

DO
    run this code
UNTIL (this condition is true)
```

```
# 
# 
# 

WH

EN
```

Returns the left two characters in the data stored in the variable `string`

Re ch st `st`

Returns the data in `string` **in upper case**

Re ch st `st`

Returns the ASCII value of the character stored in `character`

Re `st`

INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
COPYRIGHT PROTECTED
Zig Zag Education

2.2 Programming fundamentals (4) - front

Function of the code
`CHR(number)`

Function of the code
`string.substrin`
`(3,2)`

What index is the first character in a string?

Purpose of concatenation

Difference between writing to a file and appending to a file

Reading data fro a file

Description of a database field

Description of a database record

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT
PROTECTED

Zig Zag Education

2.2 Programming func

Returns two characters from the data in string, starting at the fourth character (the first character is 0)

Re
re
va

Join the values of two or more strings together into a single string

0

- Open the file in read mode
- Read the data (either line by line through a loop, or into one variable)
- Close the file

-
-

A row of data in a database table. Contains all of the fields about a person or object.

A
al
e.

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

2.2 Programming fundamentals (5) - front

Purpose and use of SQL

Function of the SQL

```
SELECT fieldName
FROM table
WHERE conditions
```

Description and features of an array

Description of 1D array

Description of 2D array

Description of array index

Searching a 1D array

Searching a 2D array

INSPECTION COPY

COPYRIGHT PROTECTED

2.2 Programming fun

- Select the field name(s) listed
- From the table named
- Where the conditions are true

1-dimensional array; a set of data values organised in a single row

- The numeric position of an item in an array
- The first element is element 0

2-
se
ca
w
co

Loop through each index in the array, e.g.

```
FOR x = 0 to 1st dimension
last element index
  FOR y = 0 to 2nd dimension
  last element index
    check array(x,y)
  NEXT y
NEXT x
```

Lo
in

```
FO
NE
```

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

2.2 Programming fundamentals (6) - front

Features of a subprogram

Features of a function

Features of a procedure

Description of a parameter

Description and example of an integer

Description and example of real

Description and example of Boolean

Description and example of a character

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

2.2 Programming fun

- Subprogram
- Self-contained code
- Called from the main program (or subprograms)
- Returns a value to where it was called from when finished
- Can take parameters from where it is called

Item of data sent to a subprogram

A decimal number, e.g. 22.22

A

A single number, letter or symbol, e.g. +

O
e.

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

2.2 Programming fundamentals (7) - front

Description of variable/constant scope

Description of local variable/consta

Description of global variable/constant

Description of random number generation

Purpose of assignment

What would `random(3,10)` **return?**

Description of string slicing

Function of the code `string.substring (start, quantity)`

INSPECTION COPY

COPYRIGHT PROTECTED

2.2 Programming fun

A variable/constant declared within a sub/main program.

The variable/constant can only be accessed in that sub/main program.

To access it elsewhere it needs passing as a parameter, or returned as a value.

This saves memory as the space is freed up when that sub/main program stops running.

A function that returns a random number between two specified values

A value between 3 and 10 (inclusive)

Starting at character start, in the `string`, return `quantity` number of characters

Va
be
in
ar
(a
th

A
ou
Th
ac
pr
Th
th
pr

To
m
va
in
Fo
X

Se
ch

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

2.2 Programming fundamentals (8) - front

Description and example of a string

Purpose of casting

Examples of arithmetic operators

Examples of Boolean operato

Identify the operator:

+

Identify the operator:

–

Identify the operator:

*

Identify the operator:

/

INSPECTION COPY

COPYRIGHT PROTECTED

2.2 Programming func

Turning data of one data type into another data type, e.g. the string "22" into the integer 22

A
e.

INSPECTION COPY

Used for comparisons; for example, in IF statements, e.g. AND, OR, NOT

U
m
e.

INSPECTION COPY

Subtraction

A

INSPECTION COPY

Division

M

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

2.2 Programing fundamentals (9) - front

Identify the operator:

`^`

Identify the operator:

`MOD`

Identify the operator:

`DIV`

Output from:

`10 MOD 3`

Output from:

`25 MOD 7`

Output from:

`99 MOD 21`

Output from:

`25 DIV 7`

Output from:

`99 DIV 21`

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

2.2 Programming fun

Modulus division returns the remainder

Ex

10 / 3 = 3 remainder 1

10 MOD 3 = 1

In

th

th

99 / 21 = 4 remainder 15

99 MOD 21 = 15

25

25

99 / 21 = 4 remainder 15

99 DIV 21 = 4

25

25

INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
COPYRIGHT PROTECTED
Zig Zag Education

2.2 Programing fundamentals (10) - front

Function of

AND

Function of

OR

Identify the comparison operator

>=

Identify the comparison operator

==

Identify the comparison operator

!=

Identify the comparison operator

<

Identify the comparison operator

>

Identify the comparison operator

<=

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

2.2 Programming funda

If one side is true the result is true

If re

Equal to

G

Less than

N

Less than or equal to

G

2.3 Producing robust programs (1) - front

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

Purpose of defensive design

Methods of defensive desig

Purpose of validation

Purpose of leng check

Purpose of range check

Purpose of existence check

Purpose of format check

Purpose of type check

INSPECTION COPY

COPYRIGHT PROTECTED

2.3 Producing robust

- Validation
- Planning for contingencies
- Anticipating misuse
- Authentication

M
ca
da
be
w

-
-

Make sure the number of characters does not exceed the limit

Make sure the data is the same as that which already exists in the system (e.g. choose from a set of options)

M
w

Make sure data is of the correct data type, e.g. string

M
er
fo

INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
COPYRIGHT PROTECTED
Zig Zag Education

2.3 Producing robust programs (2) - front

Purpose of authentication (defensive design)

Purpose of planning for contingencies

How to anticipate misuse

Purpose of system maintenance

Purpose and examples of maintainability

Description and purpose of code comments

Description and purpose of code indentation

Purpose of testi

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

2.3 Producing robust

Plan what happens if something goes wrong, e.g. an error occurs or unexpected data is input, and how data will be recovered

On
co
pa
sy

Keeping a system running, e.g. correct errors, change its function, make it more efficient

Lo
do
e.g
th
If
re

Text added to a program, which is not run, that explains the function of different sections of code

-
-

- To make sure a system works as intended
- To make sure a system is robust (cannot be easily broken)

M
co
sp
cl
in

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

2.3 Producing robust programs (3) - front

Types of testing

Purpose of testing

Purpose of final testing

Purpose of iterative testing

Three types of test data

Description of borderline test data

Description of normal test data

Description of invalid test data

INSPECTION COPY

COPYRIGHT PROTECTED

2.3 Producing robust

- To make sure a system works as intended
- To make sure a system is robust (cannot be easily broken)

Testing during the creation of the program to check that the code that has been programmed works

Te
is
m
fu

Data on the edge of what should be accepted

Data that should be rejected

Da
ac

INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
COPYRIGHT PROTECTED
Zig Zag Education

2.3 Producing robust programs (4) - front

Two types of error

Description of a syntax error

Description of a logic error

Description of a test plan

Description of refining algorithms

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

2.3 Producing robust

The program does not follow the rules of the language.

INSPECTION COPY

A formal document that lists all the ways a program needs to be tested, including the expected results. This should include normal, borderline and invalid test data.

INSPECTION COPY

**Ch
to
er
fe**

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

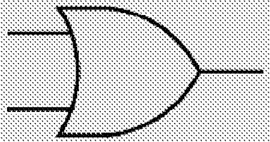
2.4 Boolean logic (1) - front

AND logic gate symbol

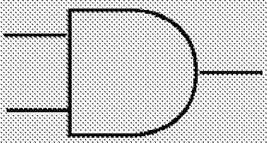
NOT logic gate symbol

OR logic gate symbol

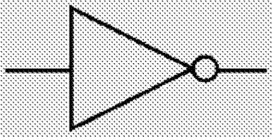
Description of



Description of



Description of



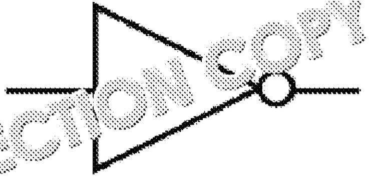
OR truth table with inputs A and B

AND truth table with inputs A and B

INSPECTION COPY

COPYRIGHT PROTECTED

INSPECTION COPY



Zig Zag Education INSPECTION COPY

- OR gate
- If one side is true, the result is true

Zig Zag Education INSPECTION COPY

- NOT gate
- If the statement is true, the result is false
- If the statement is false, the result is true

COPYRIGHT PROTECTED

Zig Zag Education

A	B	A AND B
0	[illegible]	0
0	1	0
1	0	0
1	1	1

Zig Zag Education INSPECTION COPY

2.4 Boolean logic (2) - front

NOT truth table with input A

Logic circuit for

(A AND B) OR NOT

Logic circuit for

A AND NOT(B OR C)

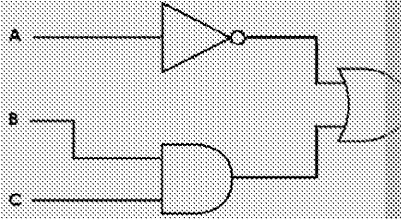
Truth table for

(A AND B) OR NOT

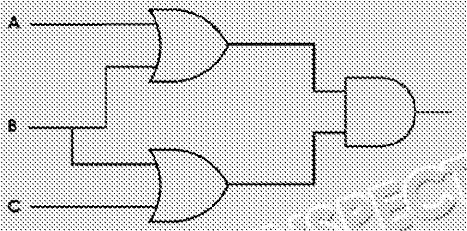
Truth table for

A AND NOT(B OR C)

Expression for



Expression for



INSPECTION COPY

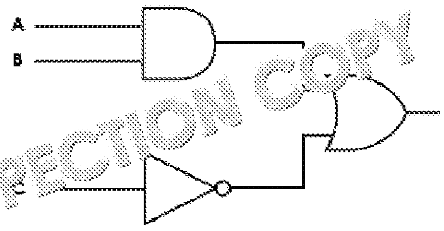
INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

2.4 Boo



A	B	C	A AND B	NOT C	(A AND B) OR NOT C
0	0	0	0	1	1
0	0	1	0	0	0
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	0	1	1
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	1	0	1

NOT A OR (B AND C)

A
0
0
0
0
1
1
1
1

(A

INSPECTION COPY

COPYRIGHT PROTECTED

2.5 Programming languages and IDEs (1) - fron

Features of a low-level language

Features of a high-level language

Benefits of a low-level language

Drawbacks of a low-level language

Benefits of a high-level language

Purpose and types of translator

Purpose and features of a compiler

Benefit of a compiler

INSPECTION COPY

COPYRIGHT PROTECTED

2.5 Programming languages

- Uses human-readable words / commands
- Must be converted into low-level language before it can be run
- One high-level command may become many low-level commands

- Difficult for people to code/understand
- Programmer needs knowledge of how memory is managed on the computer
- Program is machine dependent

- Software that converts one programming language into another, e.g. high-level to low-level
- Compiler, interpreter

Produces an executable file; this can be run without the need for a compiler/IDE, and users will not have access to the source code

INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
INSPECTION COPY
COPYRIGHT PROTECTED

2.5 Programming languages and IDEs (2) - fro

Drawback of a compiler

Purpose and features of an interpreter

Benefit of an interpreter

Drawbacks of ar interpreter

Description and features of an IDE

Features of a co editor in an IDE

Description of run-time environment in an IDE

Features of erro diagnostics in a IDE

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

INSPECTION COPY

COPYRIGHT PROTECTED

Zig Zag Education

2.5 Programming languages

- Converts the high-level code into low-level code
- Checks a line of code then runs it
- Reports errors when they are found and stops for the error to be corrected

H
ti
ta

- Needs to be reinterpreted every time before the program is run
- User will need the interpreter software to run the program

U
tr
ru
co

- Interface for writing program code
- Includes features such as auto-complete, auto-indent and syntax highlighting

- Allows user to test the code
- Features include error reports; variable watch window (for monitoring variable values); breakpoint (for pausing at a precise point for debugging); step-through (for running the code line by line)

R
di
us

INSPECTION COPY

COPYRIGHT PROTECTED