Data Representation

A-Level Computer Science · Vocabulary in context

Word	Example
user-defined type /ˈjuːzə dɪˈfaɪnd taɪp/	A **user-defined type** is a type the programmer makes.
field /fiːld/	The **field** “Name” stores each student’s name. ”Name”
record /ˈrekɔːd/	Each **record** in the school database holds one student’s details.
set /set/	A **set** holds values with no repeats.
class /klæs/	A **class** is a plan for making objects.
composite type /ˈkɒmpəzɪt taɪp/	A record is a **composite type**: it holds several values.
enumerated type /ɪˈnjuːməreɪtɪd taɪp/	The **enumerated type** Day holds Monday to Sunday. Day Monday Sunday
pointer /ˈpɔɪntə/	The **pointer** shows where the next item is.
linked list /lɪŋkt lɪst/	In a **linked list**, each item points to the next one.
dereference /ˌdiːˈrefrəns/	**Dereference** the pointer to get the value it points to.
object /ˈɒbdʒekt/	Each car in the game is an **object**.
attributes /ˈætrɪbjuːts/	A car’s **attributes** include its colour and speed.
methods /ˈmeθədz/	A car’s **methods** include Drive() and Stop(). Drive() Stop()
constructor /kənˈstrʌktə/	The **constructor** sets up a new object.

exercises.lol exercises.com.cn

Word	Example
File organisation /faɪl ˌɔːgənaɪˈzeɪʃn/	**File organisation** is how the records are arranged in a file.
serial file /ˈsɪərɪəl faɪl/	A **serial file** stores records in the order they arrive.
sequential file /siːˈkwenʃl faɪl/	A **sequential file** stores records in key order.
random file /ˈrændəm faɪl/	In a **random file**, a hash decides where each record goes.
direct access /daɪˈrekt ˈækses/	**Direct access** goes straight to the record you want.
hash function /hæʃ ˈfʌŋkʃn/	A **hash function** turns a key into an address.
sequential access /siːˈkwenʃl ˈækses/	**Sequential access** reads the records one by one.
deterministic /dɪˌtɜːmɪˈnɪstɪk/	A **deterministic** algorithm always gives the same output for the same input.
collision /kəˈlɪʒn/	A **collision** happens when two computers send at the same time.
linear probing /ˈlɪnɪə ˈprəʊbɪŋ/	**Linear probing** tries the next space when one is full.
chaining /ˈtʃeɪnɪŋ/	**Chaining** keeps a list at each address.
load factor /ləʊd ˈfæktə/	A high **load factor** means the table is nearly full.
overflow area /ˌəʊvəˈfləʊ ˈeərɪə/	Extra records go into the **overflow area**.
overflow /ˌəʊvəˈfləʊ/	Adding 200 and 100 in one byte causes **overflow**. 200 100

exercises.lol exercises.com.cn

Word	Example
• **floating-point** /ˈfləʊtɪŋ pɔɪnt/	**Floating-point** numbers can store very large and very small values.
mantissa /mænˈtɪsə/	The **mantissa** holds the digits of the number.
exponent /ekˈspəʊnənt/	The **exponent** moves the binary point.
two's complement /tuːz ˈkɒmplɪmənt/	In **two's complement**, 11111111 means −1. 11111111 −1
normalised /ˈnɔːməlaɪzd/	A **normalised** number starts with 01 or 10. 01 10
rounding errors /ˈraʊndɪŋ ˈerəz/	**Rounding errors** make 0.1 + 0.2 not exactly 0.3. 0.1 + 0.2 0.3
underflow /ˌʌndəˈfləʊ/	Popping from an empty stack causes **underflow**.
fixed-point /fɪkst pɔɪnt/	In **fixed-point**, the binary point never moves.
BCD /ˌbiː siː ˈdiː/	In **BCD**, each denary digit gets its own four bits.
precision /prɪˈsɪʒn/	More mantissa bits give more **precision**.
range /reɪndʒ/	More exponent bits give a bigger **range**.

exercises.lol exercises.com.cn