SKILLSEA

Module 6 Computer Science

6.2 Data types, operators and codes

Co-funded by the Erasmus+ Programme of the European Union

Introduction

- Explain the meaning of and provide examples of Data types
- Explain how bits can be combined to create complex codes
- Introduce the concept of Boolean operators
- Differentiate between a digital signal, machine language, assembly languages and high level languages.

Co-funded by the Erasmus+ Programme of the European Union

Learning Outcomes

- 6.2. Understanding computer binary operations and introduces terms Data type, Machine Code, Assembly Language. Identifies the relationship between and use of common character sets.
  - At the completion of the session, students will be able to;
  - 1. Explain the meaning of the term data type and provide examples
  - 2. Describe the representation of a Bit in various forms
  - 3. Identify basic Boolean (logical) operators AND, OR, NOT, XOR
  - 4. Explain the use of binary codes to represent characters
  - 5. Explain the term ‘character set’
  - 6. Describe with examples (for example ASCII and Unicode) the relationship between the number of bits per character in a character set and the number of characters which can be represented

Co-funded by the Erasmus+ Programme of the European Union

Recalling

- How is data represented in a computer or digital system?
- BITS.
- 1 – 0
- On- 0ff
- True-False
- 5volts 0 volts.

- Each bit can be grouped' and combined to make codes.

Data Types

- A data type, in computer science and programming, is a classification that specifies which type of value a **variable** has and what type of **mathematical, relational or logical operations** can be applied to it without causing an error.
- The data type defines which operations can safely be performed to **create, transform and use the variable in another computation.**

- **Question – what data type did we identify in Lesson 6.1?**

Examples of some common data types

Data Type	Used for	Example
Boolean	Representing logical values	TRUE, FALSE
Character	Encoding text numerically	97 (in ASCII, 97 is a lower case 'a')
String	Alphanumeric characters The digits are treated as text.	m.v Jolly Sailor, GBTT, MMSI 235123456
Integer	Whole numbers have numeric value	7, 12, 999
Float (floating point)	Number with a decimal point with numeric value	3.15, 9.06, 00.13
There are many other data types which you will encounter. For example Array, Date, Time, Datetime, Timestamp etc. Commonly encountered in ships equipment such as GNSS, ECDIS Data Loggers		

Why data types are important

- A data type is an attribute associated with a piece of data that tells a computer system **how to interpret its value.**
- Understanding data types ensures that data is collected in the preferred format and the value of each property is as expected.
- For example, knowing the data type for “Jolly Sailor” will help a computer know:
  - whether the data is referring to the ship name (“Jolly Sailor”) or a list of two names (“Jolly” and “Sailor”)
- Understanding data types will help you ensure that:
  - the data collected is always in the right format (“Jolly Sailor” vs. “Sailor,Jolly”)
  - the value is as expected (“Joly Sailor” vs. “J011y, $ai110r”)

Co-funded by the Erasmus+ Programme of the European Union

Coding – Machine Code

- Machine code is a computer programming language comprising hexadecimal or binary instructions which computers are able to respond to directly.
- Machine code is written in a **machine language.** Therefore, a machine, i.e., a computer, can execute it without any translation or conversion.
- The instructions that exist in machine code are known as machine instructions.
- Machine code – a numerical language

Co-funded by the Erasmus+ Programme of the European Union

Coding – Assembly Language

- An assembly language is a type of **low-level programming language** that is intended to communicate directly with a computer’s hardware.
- Unlike machine code, which consists of binary and hexadecimal characters, assembly languages are designed to be readable by humans.
- Low-level programming languages such as assembly language are a necessary bridge between the underlying hardware of a computer and the higher-level programming languages—such as Python or JavaScript—in which modern software programs are written.

Co-funded by the Erasmus+ Programme of the European Union

Recap Question

- How are these Data types processed by the computer?
- As bits.

Boolean and Logical operators

- Switches can be combined to
- AND
- OR
- NOT
- XOR

AND

- 0 AND 0 = 0
- 1 AND 0 = 0
- 0 AND 1 = 0
- 1 AND 1 = 1
- Described in a TRUTH TABLE

- Only when both states are 1, is it true that the Output is '1'

OR

- 0 OR 0 = 0
- 1 OR 0 = 1
- 0 OR 1 = ?
- 1 OR 1 = ?

NOT

- NOT 0 = 1
- NOT 1 = 0

XOR

- A special form of OR function

- 0 OR 0 = 0
- 1 OR 0 = 1
- 0 OR 1 = 1
- **1 OR 1 = 0**

Character Set

- What is meant by a 'character set'.
- A defined list of characters recognised by the computer hardware and software.
- Each character is represented by a number.

- Which Character Set was introduced in Lesson 6.1 ?

ASCII

- Lesson 6.1. introduced the ASCII character set.
- ASCII - **A**merican **S**ystem **C**ode for **I**nformation **I**nterchange
- Developed for "Teletype" and "Teleprinters" in 1963
- ASCII Traditionally used 7 bits,
- 00000000 to 11111111
- How many states can be represented?

TASK

- Calculate the maximum number of characters that can be represented using 7 bits.

2^n

- Each Bit can only have one of two states – hence the term Binary.
- 2^0 = 1 state
- 2^7 = 128 states
- So 7 bits can represent 128 different codes
- 7 bit ASCII
- The original ASCII character code, which provides 128 different characters, numbered 0 to 127. ASCII and 7-bit ASCII are synonymous.

ASCII 7 bit Code

- What is the DECIMAL CODING FOR ASCII ?

- 65, 83, 67, 73,73

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(	72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29	)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-_	63	3F	?	95	5F	_	127	7F	DEL

Typical shipboard example

- Ships NAVTEX
- Navigational Telex
- Based on the original Teleprinter designs
- Receives Navigational Safety and weather information.

8 Bit ASCII

- ASCII was originally developed for basic computers and printers such as the NAVTEX
- As more computers began to work with 8-bit groups of data, ASCII was written as 8 bits.
- In NAVTEX for example, some ERROR CHECKING is required.
- The most significant bit is sometimes used as a **parity bit** to perform a parity check (a form of error checking).

ASCII – 8 bits

- 2^8 = 256 states
- Required

Hexadecimal Representation

- What are the alternative HEX Codes for the Characters
- 'A' 'S' 'C' 'I' 'I'

UNICODE

- Uses 16 bits
- How many codes are available?
- 2^{16}
- Explore UNICODE using the link
- https://unicode-table.com/en/#0089

Summary

- Developed a better understanding of codes
- Considered Logical Functions.
- Used the ASCII Code
- Reviewed number bases of 2, 16 BINARY and HEXADECIMAL
- Introduced UNICODE
- Lesson 6.3 develops an understanding of basic structure of a computer system

Co-funded by the Erasmus+ Programme of the European Union

Activity

- How many bits are used for the ASCII code?
- How many characters will this number of bits allow for?
- From what was the ASCII code developed?
- What is extended ASCII?
- Why was Unicode developed?

Co-funded by the Erasmus+ Programme of the European Union

Self assessment quiz

- Note – develop an online interactive quiz or multiple choice selection.
- Example follows

Co-funded by the Erasmus+ Programme of the European Union

Suggestive Self Assessment

- (a) Explain the meaning of Data Type
- (b) What data types would represent the values “Fuel Oil”, 308.5 , “Tonnes”, YES or NO
- (c) Describe the primary difference between Machine Code and Assemble Language
- (d) Explain what is meant by a ‘character set’.
- (e) ASCII is an acronym. State what words the initial letters represent.
- (f) The first edition of the ASCII code was published in 1963 and it was based on an earlier code.
  - (i) Why was it developed.
  - (ii) Give a marine example of where it is used.

Co-funded by the Erasmus+ Programme of the European Union

Suggestive Self Assessment

- (g)
  - (i) How many bits are used for this original code?
  - (ii) How many characters can be encoded using this number of bits?
- (h) Not all of the characters are printable. What are the others called and what are they used for?
- (i) Explain how ASCII is used to represent text in a computer system.
- (j) Unicode is also used to represent text in a computer system.
- (k) Explain the difference between the character sets of Unicode and ASCII.

Co-funded by the Erasmus+ Programme of the European Union