IT
Study guideline for grade 10

Written exam includes two parts:

1st part: (20 marks)

Primitive data types and converting strings to their hexadecimal values using ASCII table.

2nd part measures your practical skills in python programming.

General questions about python and write code based on what you have learned.

PRIMITIVE DATA TYPES AND THEIR STORAGE

The primitive data types used when programming are described in the table below.

Table 2.5: *The primitive data type*

NAME	DESCRIPTION	REPRESENTATION	EXAMPLE
Boolean	A data type that can only be TRUE or FALSE.	Boolean values can be represented by a single bit of data that is either ON or OFF.	TRUE or FALSE 1 or 0 YES or NO
Integer	An integer is any positive or negative whole number.	Any integer up to 255 can be represented by one byte of data. Larger integers require more bytes of data.	1 or 82355 or –59
Float	A float refers to any number with a decimal value.	To represent floats require multiple bytes of data.	82.355 or 0.55 or –10.02
Char	A char is a data type that can store a single character. This includes letters, numbers, and symbols.	Each char requires one byte (8 bits) of data.	'A' or 5 or '1' or '@'

NAME	DESCRIPTION	REPRESENTATION	EXAMPLE
String	A string refers to several characters, letters, or numbers. Strings are always surrounded by quotation marks to show where they begin and end. NB: If numbers are represented as a string, you cannot use them in your calculations.	Each character in a string requires one byte of data.	Tel: 082 111 2222 'Hello, World!' 'Maria Messina'

Activity 2.4

2.4.1 Boolean refers to:

a. A data type that can only be TRUE or FALSE.

b. Any positive or negative whole number.

c. A data type that can store a single character.

2.4.2 A float is represented by:

a. A single bit of data

b. One byte of data

c. Multiple bytes of data

2.4.3 Which data types refer to several characters, letters, or numbers?

a. Char

b. String

c. Float

2.4.4 Which of the following is an example of a Char?

a. '!'

b. True/False

c. 'Hello World!'

2.4.5 The terms 'information' and 'data' are often used in IT. Which description best fits these terms?

a. Information is understood by non-technological people, on the other hand ASCII code is data that can only be read and understood by IT professionals.

b. Information is the result of data that has been processed into something that is meaningful.

c. Data is the result of information *supplied by the user*.

d. Data and information are the same; it only depends on the device that it is created for.

2.4.6 Write down the data type that would be used to store each of the following items:

a. '1'

b. FALSE

c. 'Perhaps'

d. 85

e. 75,299

f. '43,0'

g. 'True'

h. 'a'

i. 0,851

j. 4951327

2.4.7 Write down the correct primitive data type for each item described:

a. The average of the marks that a class of 21 learners obtained in a test.

b. The answer to the question: Do you like chocolates?

c. The millilitres of milk required in a cake recipe.

d. A cell phone number.

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	*[NULL]*	32	20	*[SPACE]*	64	40	@	96	60	`
1	1	*[START OF HEADING]*	33	21	!	65	41	A	97	61	a
2	2	*[START OF TEXT]*	34	22	"	66	42	B	98	62	b
3	3	*[END OF TEXT]*	35	23	#	67	43	C	99	63	c
4	4	*[END OF TRANSMISSION]*	36	24	$	68	44	D	100	64	d
5	5	*[ENQUIRY]*	37	25	%	69	45	E	101	65	e
6	6	*[ACKNOWLEDGE]*	38	26	&	70	46	F	102	66	f
7	7	*[BELL]*	39	27	'	71	47	G	103	67	g
8	8	*[BACKSPACE]*	40	28	(	72	48	H	104	68	h
9	9	*[HORIZONTAL TAB]*	41	29	)	73	49	I	105	69	i
10	A	*[LINE FEED]*	42	2A	*	74	4A	J	106	6A	j
11	B	*[VERTICAL TAB]*	43	2B	+	75	4B	K	107	6B	k
12	C	*[FORM FEED]*	44	2C	,	76	4C	L	108	6C	l
13	D	*[CARRIAGE RETURN]*	45	2D	-	77	4D	M	109	6D	m
14	E	*[SHIFT OUT]*	46	2E	.	78	4E	N	110	6E	n
15	F	*[SHIFT IN]*	47	2F	/	79	4F	O	111	6F	o
16	10	*[DATA LINK ESCAPE]*	48	30	0	80	50	P	112	70	p
17	11	*[DEVICE CONTROL 1]*	49	31	1	81	51	Q	113	71	q
18	12	*[DEVICE CONTROL 2]*	50	32	2	82	52	R	114	72	r
19	13	*[DEVICE CONTROL 3]*	51	33	3	83	53	S	115	73	s
20	14	*[DEVICE CONTROL 4]*	52	34	4	84	54	T	116	74	t
21	15	*[NEGATIVE ACKNOWLEDGE]*	53	35	5	85	55	U	117	75	u
22	16	*[SYNCHRONOUS IDLE]*	54	36	6	86	56	V	118	76	v
23	17	*[END OF TRANS. BLOCK]*	55	37	7	87	57	W	119	77	w
24	18	*[CANCEL]*	56	38	8	88	58	X	120	78	x
25	19	*[END OF MEDIUM]*	57	39	9	89	59	Y	121	79	y
26	1A	*[SUBSTITUTE]*	58	3A	:	90	5A	Z	122	7A	z
27	1B	*[ESCAPE]*	59	3B	;	91	5B	[	123	7B	{
28	1C	*[FILE SEPARATOR]*	60	3C	<	92	5C	\	124	7C	|
29	1D	*[GROUP SEPARATOR]*	61	3D	=	93	5D	]	125	7D	}
30	1E	*[RECORD SEPARATOR]*	62	3E	>	94	5E	^	126	7E	~
31	1F	*[UNIT SEPARATOR]*	63	3F	?	95	5F	_	127	7F	*[DEL]*

Python

Rose Barakat

What is Python?

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

Defining Python

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.

IDE

Thonny

File Extention

.py

Exercise:

creat simple python program in thonny and save it in the computer then see the extention of the file

```
print("Hello, World!")
```

Python Indentation

Indentation refers to the spaces at the beginning of a code line.
Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important.
Python uses indentation to indicate a block of code.

Python Indentation

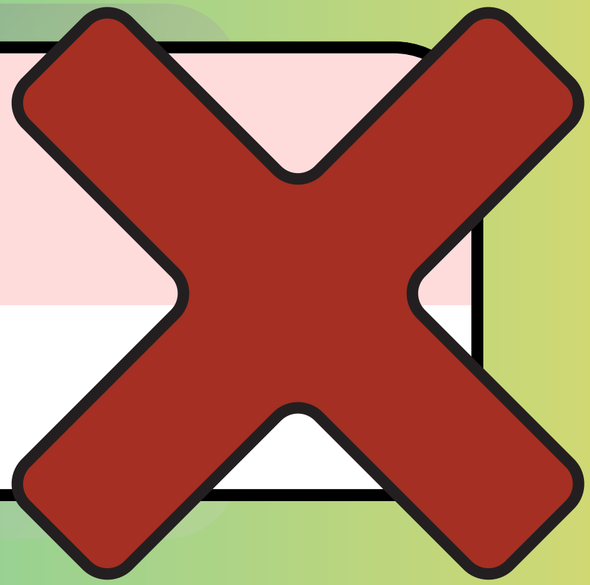
Example

```
if 5 > 2:
  print("Five is greater than two!")
```

Example

Syntax Error:

```
if 5 > 2:
print("Five is greater than two!")
```



Python Indentation

The number of spaces is up to you as a programmer, the most common use is four, but it has to be at least one.

Example

```
if 5 > 2:
 print("Five is greater than two!")
if 5 > 2:
        print("Five is greater than two!")
```

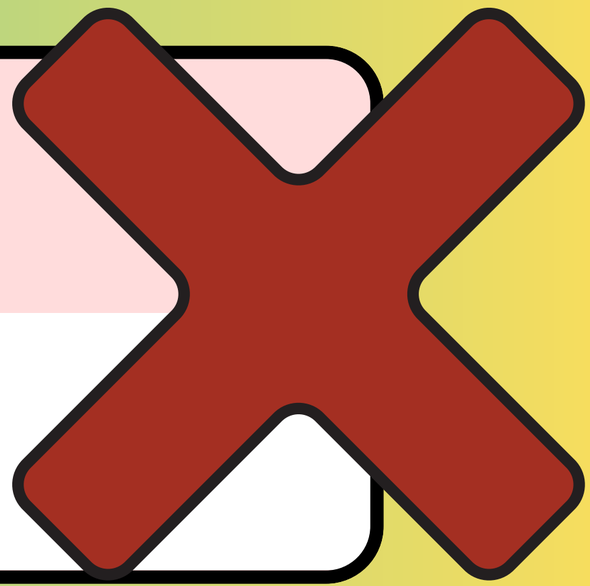
Python Indentation

You have to use the same number of spaces in the same block of code, otherwise Python will give you an error:

Example

Syntax Error:

```
if 5 > 2:
 print("Five is greater than two!")
        print("Five is greater than two!")
```



Python Variables

Example

Variables in Python:

```
x = 5
y = "Hello, World!"
```

In Python, variables are created when you assign a value to it:

Python Variables

Example

```
x = 4
x = "Sally"
print(x)
```

Variables do not need to be declared with any particular type, and can even change type after they have been set.

Comments

Example

Comments in Python:

```
#This is a comment.
print("Hello, World!")
```

Comments start with a #, and Python will render the rest of the line as a comment:

Casting

If you want to specify the data type of a variable, this can be done with casting.

Example

```
x = str(3)    # x will be '3'
y = int(3)    # y will be 3
z = float(3)  # z will be 3.0
```

Apply this on Thonny, then print each value.

Get the Type

You can get the data type of a variable with the type() function.

Example

```
x = 5
y = "John"
print(type(x))
print(type(y))
```

Apply this on Thonny.

Single or Double Quotes?

String variables can be declared either by using single or double quotes:

Example

```
x = "John"
# is the same as
x = 'John'
```

Case-Sensitive

Variable names are case-sensitive.

Example

This will create two variables:

```
a = 4
A = "Sally"
#A will not overwrite a
```

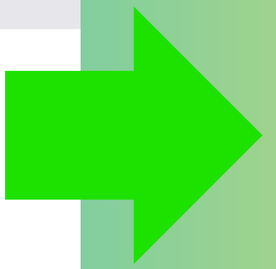
Python Booleans

Booleans represent one of two values: True or False.

In programming you often need to know if an expression is True or False.
You can evaluate any expression in Python, and get one of two answers, True or False.
When you compare two values, the expression is evaluated and Python returns the Boolean answer:

Example

```
print(10 > 9)
print(10 == 9)
print(10 < 9)
```



it will print:

True
False
False

Example

Print a message based on whether the condition is `True` or `False`:

```
a = 200
b = 33

if b > a:
  print("b is greater than a")
else:
  print("b is not greater than a")
```

Any string is True, except empty strings.
Any number is True, except 0.
Any list, tuple, set, and dictionary are True, except empty ones.

The following will return True:

```
bool("abc")
bool(123)
bool(["apple", "cherry", "banana"])
```

The following will return False:

```
bool(False)
bool(None)
bool(0)
bool("")
bool(())
bool([])
bool({})
```