1. The general syntax of turtlescript is:
[TYPE]@[VARIABLE NAME] = [VALUE];
the same could be written as:
[TYPE]@[VARIABLE NAME]= [VALUE];
2. One type (in the above code [TYPE]) is:

ID	**TYPE**	**C++ Typ**
1	INTEGER	TUR_INTEGER
2	BOOLEAN	TUR_BOOLEAN
3	STRING	TUR_STRING
4	BIGNUM	TUR_BIGNUM
5	BYTE	TUR_BYTE
6	FLOAT	TUR_FLOAT

3. *INTEGER*
An integer is stored as TUR_INTEGER (int).
It has a value range of -2147483648 to 2147483647.
The minus sign is written directly in front of the number.
4. *BOOLEAN*
A Boolean is stored as TUR_BOOLEAN (int).
It can either take the value true(true) or false(false).
5. *STRING*
A string is saved as TUR_STRING (string).
The text of the string is written in quotation marks..
You can use the double quotation mark (").
6. *BIGNUM*
A Bignum is saved as TUR_BIGNUM (long long).
It has a range of values from -(2^{63}) to(2^{63})-1.
The minus sign is written directly in front of the number.
7. *BYTE*
One byte is stored as TUR_BYTE (unsigned char).
It has a range of values from 0 to 255.
It should be avoided to store higher values than 255 in one byte.
The range of values of a byte is so large that the ASCII alphabet (also ASCII code) can be tuned.
8. *FLOAT* A float is saved as TUR_FLOAT (string).
In this case, the float is stored as a string until the evaluation or read-out of the data.
Only when it is read, it is converted into a float.

9.(C++) To compile the files with a C ++ compiler you need the C++ Boost Library.
10.(C++) In C ++, there will be a namespace turtlescript successfully including the header file and successfully compiling the turtlescript files.
In this namespace are the functions editvar, eval, context, stack, getinteger, getboolean, getstring, getbignum, getbyte, getfloat and clear.
11.(C++) *editvar*
Not intended for the user / developer.
12.(C++) *eval*
Interpret turtlescript code.
As the first argument, request the turtlescript code, which should be interpreted.
Always returns one.
13.(C++) *context*
Only partially intended for the user / developer.
Interpret turtlescript code value.

As the first argument, demand the value that should be interpreted.
Returns the evaluated value.
Value refers to the assignment of the code after the assignment (=).
14.(C++) *stack*
Returns the currently used stack as TUR_STACK.
15.(C++) *getinteger*
Converts a variable to type TUR_INTEGER.
The first argument is the variable name of the variable from which the content is to be converted.
16.(C++) *getboolean*
Converts a variable to type TUR_BOOLEAN.
The first argument is the variable name of the variable from which the content is to be converted.
17.(C++) *getstring*
Converts a variable to type TUR_STRING.
The first argument is the variable name of the variable from which the content is to be converted.
18.(C++) *getbignum*
Converts a variable to type TUR_BIGNUM.
The first argument is the variable name of the variable from which the content is to be converted.
19.(C++) *getbyte*
Converts a variable to type TUR_BYTE.
The first argument is the variable name of the variable from which the content is to be converted.
20.(C++) *getfloat*
Converts a variable to type TUR_FLOAT.
The first argument is the variable name of the variable from which the content is to be converted.
21.(C++) *clear*
Empty the variable stack.
Always returns one.
22. *Stack*
Variables in turtlescript are stored in a stack of type TUR_STACK.
(C++) A TUR_STACK is a struct containing three vectors.
In the first vector, named name and the storage type string, the names of the variables are stored.
In the second vector, with the name value and the memory type any (from Boost), the contents of the variables are stored.
In the third vector, with the type and the memory type int, the types (see point 2) of the variables are stored.

23.(C++) *TUR_EXCEPTION*
The TUR_EXCEPTION class in an Exception class (not derived from the exception (std) class) that is thrown when an interpretation error occurs.
It contains the function what which requests nothing as an argument and returns the error message.
It contains the function line which requests nothing as argument and returns line number in which error has happened.
24. To assign the value of one variable to another, the dollar sign ($) is used.
Behind the dollar sign ($) you write the variable name, from which the content should be displayed.

Arithmetic operations, such as +, -, *, / or similar, are not possible.
The chaining of strings is also not possible.
25. A turtlescript file has the file extension .tur or .turtle and is usually stored with the UTF-8 encoding.