C++

PRINTING

std::cout << "Hello World!\n";

•Also known as print.

•add ; every line.

COMPILING AND EXECUTING

g++ {File Name}

•Compiling the code from the code editor to be understood by the computer

./a.out

•Executes said code

•Both to be written on the black terminal

g++ {File Name} -o {Nickname}

•gives the file a specific name

./{Nickname}

•Executes said file

COMMENTS

Single-line comment

// {comment} //

Multi-line comment

/\* {comment} \*/

VARIABLES

Types:

int - integer or numbers

double - 15 digit decimal

Float - 7 digit decimal

char - a single character or letter

string - a sequence of characters

bool - true or false value

int (operators):

+, -, \*, /, % (modulo, divides and gives the remainder)

pow({variable name}, {power})

making a variable

{type of variable} {name of variable}; or

{type of variable} {name of variable} = {value of variable};

•value of variable can be indicated later

std::cout << score << "\n";

•to check the value of the variable

CHAINING

use << to add another line of string or connect them with variables

USER INPUT

std::cin >> {variable name}

CONSTANT

const {variable}

•cannot be changed

TYPE CONVERSION

{variable1 type} {variable1 name};

{variable2 type} {variable2 name};

{variable2 name} = (new variable1 type) {variable1 name};

RELATIONAL OPERATORS

== (equal), != (not equal), >, <, >=, <=

•Used for comparisons or conditioning

• = is used to give a variable a value

CONDITIONAL STATEMENTS

if (condition) { - condition

else if (condition) { - multiple conditions

else (condition) { - if condition is not accomplished

SWITCH STATEMENTS

switch{(variable name)} { - the value of {variable name} will be compared with the following case statements

case {value} : - if the value of the {variable name} is equal to the case, it will execute its program

default : - if there was no case statements that was equal to the value of the {variable name} the it will execute the default program

LOGICAL OPERATORS

if (variable1 && variable2) { - the and logical operator

|| (variable1 || variable2) { - the or logical operator

! (!variable) { - the not logical operator

•These will result in either true or false

LOOPS

for {(counter); (continue condition); (incrementals ie. (variable)++ or --)} { - will continue until the (continue condition) is completed or the (counter) is out

while {(condition)} { - can be continued until the (condition) is fulfilled

ERRORS

Compile-time errors: Errors found by the compiler.

Link-time errors: Errors found by the linker when it is trying to combine object files into an executable program.

Run-time errors: Errors found by checks in a running program.

Logic errors: Errors found by the programmer looking for the causes of erroneous results.

VECTORS (STORE DATA)

std::vector<{variable type}> {variable name} = {{data}} - stores data in a form of a table

std::cout << {vector name}[{index no. or data no.}] - starts from 0

{vector name}.push\_back({"data"}) - adds a new data at the front of the vector

{vector name}.pop\_back()

std::cout << {vector name}.size() << "\n"; - tells the number of data in the vector

ARRAYS (DATA CANNOT BE ADDED OR REMOVED)(SIZE CAN'T BE CHANGED)

{variable type} {variable name}[{array size}];

{variable type} {variable name}[] = {{data}};

•data can be modified or changed

FUNCTIONS (A SET OF CODES)

{return type} {function name}({parameter})

void - no return, does not give an output to the terminal

return (can be used on most kinds of return type (double, int, bool, char, std::string, and std::vector)) - returns an output

PARAMETERS AND ARGUMENT

{return type} {function name}({variable1 type} {variable1 name}, ...) {

}

{function name}({variable1 value});

•the {variable1 value} will get substituted to those with the {variable1 name} in the codes of the corresponding function.