basics

BASICS SYNTAXES IN C++

Author: Felix Okoronkwo
Date: [19/8/23]
Description: This section will be about the basic operations in c++

STATEMENTS

defination : this is the smallest independent unit of computaion in c++

```
// statement in cpp
std::cout << "Hello, World!" << std::endl;
```

most (but not all) statements in c++ end with a semicolon `;`.

FUNCTION AND MAIN FUNCTION

defination : this is a collection of statements that are executed in order sequentially top to bottom.

Every c++ program must have a main function which is the entry point of the program. the name of the function is always **main**. function perform specific jobs.

```
#include <iostream>             // preprocessor directive

int main ()                     // main function
{
  std::cout << "text";          // statement
  return 0;                     // return statement.
}
```

COMMENTS

Single line : `// this is a single line comment`
Multiline : `/** multiline comment ** /`

OBJECTS AND VARIABLES

C++ access memory for data manipulation through objects

Object : Region of storage usually memory that can store data or a value and has other associated properties.

Objects can be named or anonymous. A named object is known as a **variable** and its name its known as an **identifier**

VARIABLE INSTANTIATION, DEFINATION, TYPES, ASSIGNMENT AND ASSIGNMENT

c++ is statically typed programing languaage and this means just like in C a variable's type must be stated whenever defining a variable/object

```
int a, b;
double index;
int c; double h;             // correct but not best practice
```

Instantiation: at runtime programs are instatiated this is when objects and variables are created and assigned a memory address for future manipulation or access. variables must be instantiated bfoere the can be used to store values. an instatiated object is called an **Instance**.

Datatypes: datatypes of an obkject or variable must be known at compile time and variables onl stored the data of the same datatype as the variable itself, as stated above (**statically-typed**).

Varible structure: variables possess an **identifier**, **type** and **value**.

Variable declaration: telling the compiler that a variable exists without allocating memory for it.

Variable defination: telling the compiler that a variable exists and allocating memory for it.

Variable assignment: cassigning a value for a variable to hold in its storsge after defination.

```
extern int declared;
int defined;
defined = 0;                //assigned.
defined = 69;               // re-assignment to 69
```