Date: / / Sat. Sun. Mon. Tue. Thu. Wed. Fri. Subject: ----------

آموزش C++

The C++ programming Language (STROUSTRUP)

Chapter2: The Basics

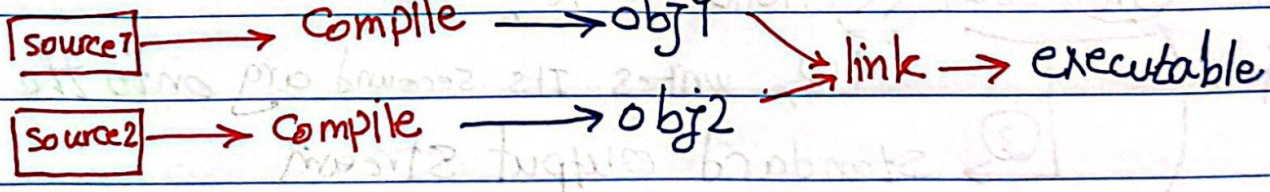
Goal: present Notation of C++, cpp's model of memory and computation, organizing code into a program

Basics:

C++ is compiled Language

many Source Texts processed by a compiler, producting obj file

obj files combine by a linker yeilding an executable



C++ portability mean is source code is portable

ISO C++ standard defines two kind of entities:

1) core language feature
2) standard library components

C++ is statically typed language ⇒ type of every entity must be known to the compiler.

Date: / / Sat. Sun. Mon. Tue. Thu. Wed. Fri. Subject: ----------

2.2.1) Hello world.

④ → return value, if no value returned, system indicate success

int main() { ① → Takes no args and does nothing

③ ② → express grouping in C++

{ each cpp project must have one global function named main

↳ None zero value from main() indicates failure

#include <iostream> ⑦ → instruct compiler to include the declaration of standard I/O facilities

```
int main(){
    std::cout<<"Hello world";
}
```

③ → standard output stream

② → writes its secound arg onto the first

① → Specifies that the name cout is to be found in standard-library namespace

note

you can use using namespace std; to make names from a namespace visible

you can go to github:
AminGhorbani96/CppStroustrup
to see an Example: ProjectName: "2.2.1"

Date: Sat Sun Mon Tue Thu Wed Fri Subject: ----------

2.2.2) Types, Variable & Arithmetic.

Every name & expression has a type.
determines operations that may be Performed on it.

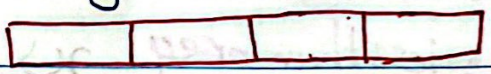
cpp has a variety of fundamental types:

bool, Char, int, double

each of them has a fixed_size

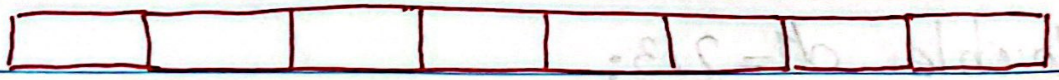
→ determines the range of values can be stored

bool : 

int: 

char: 

double:



you can obtain size of each type by sizeof() in byte:

sizeof(char) ⟹ 1

sizeof(int) ⟹ 4

Date: / / Sat. Sun. Mon. Tue. Thu. Wed. Fri. Subject: ----------

Arithmatic operations:

x+y : Plus +x: unary Plus

x-y: minus -x: unary minus

x*y: multiply x/y: devide

x%y: reminder

Comparision operations:

x==y: equal x!=y: not equal

x<y: less than x>y: greater than

x<=y: less than or eq x>=y: greater than or eq

initilizer:

```
double d1=2.3;
double d2{2.3};
```

```
complex<double> x = 1;
complex<double> x{d1, d2};
complex<double>x ={d1, d2};
```

you can use auto → we use when we don't have reson to specify type

auto x = true → x is bool

auto y = 1.2 → y is float

2.2.3. Constant

Const → I promise not to change this value

Constexpr → to be evaluated at compile time

```
const int dmv=17;
int var=17;
constexpr double max1=1.4*square(dmv); → ok
constexpr double max2=1.4*square(var); → Error
const double max3=1.4*square(var); → ok
```

2.2.4) TESTS & Loops

C++ prorides a conventional set of statements for selection & loops:

```
1) if (answer == 'y') return true;
2) switch (answer){
   case 'y':
       return true;
   case 'n':
       return false
   default:
       cout << "Something";
       return false;
```

Date: / / Sat. Sun. Mon. Tue. Thu. Wed. Fri. Subject: ----------

3) int tries=1;

while (tries<1){

⋮

tries++

}

you can go to github:
AminGhorbani96/cppstroustrup
to see full Examples of
selections & loops in Project:
"2.2.4"

4)