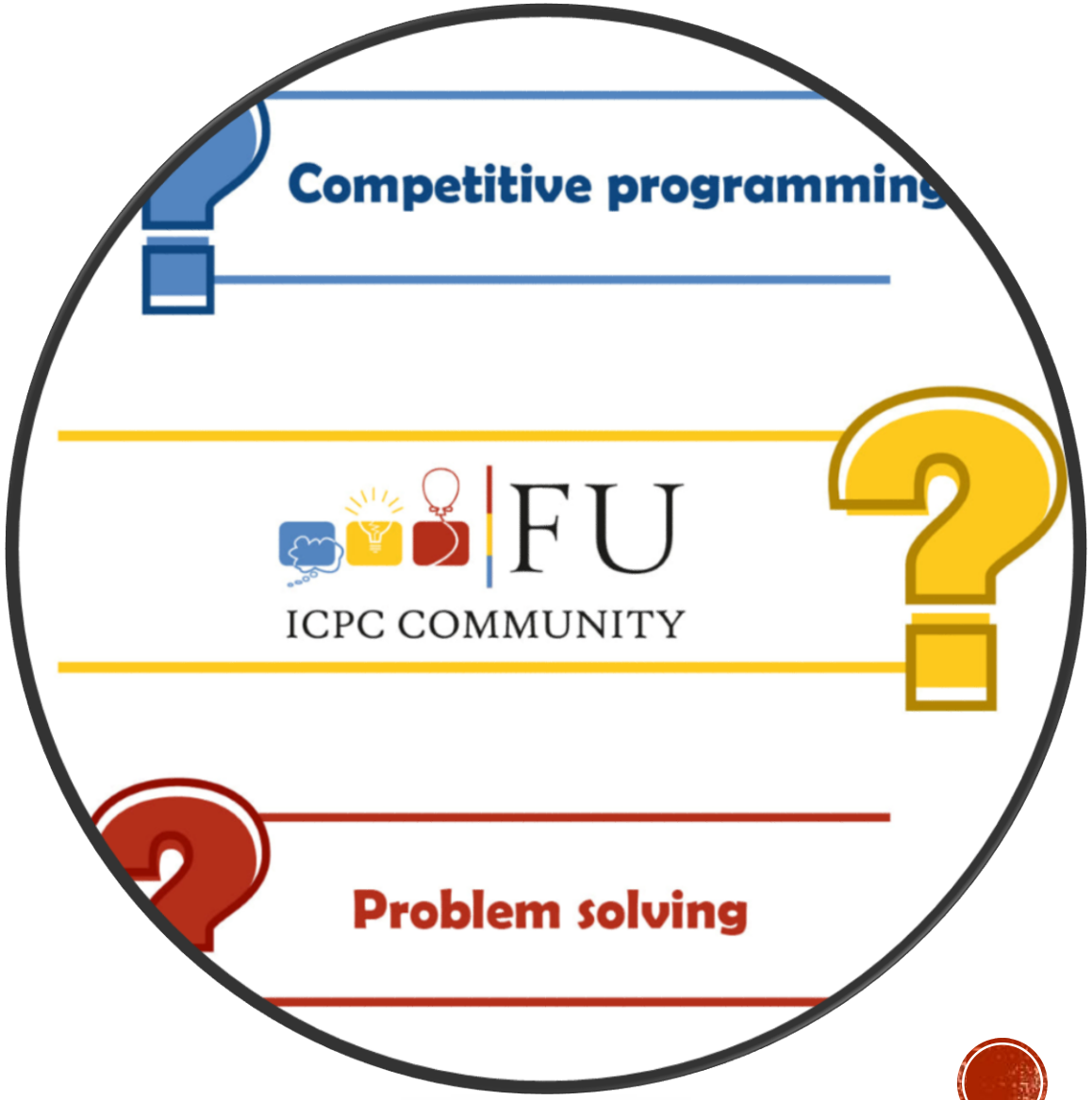
Intro C++

SESSION 1

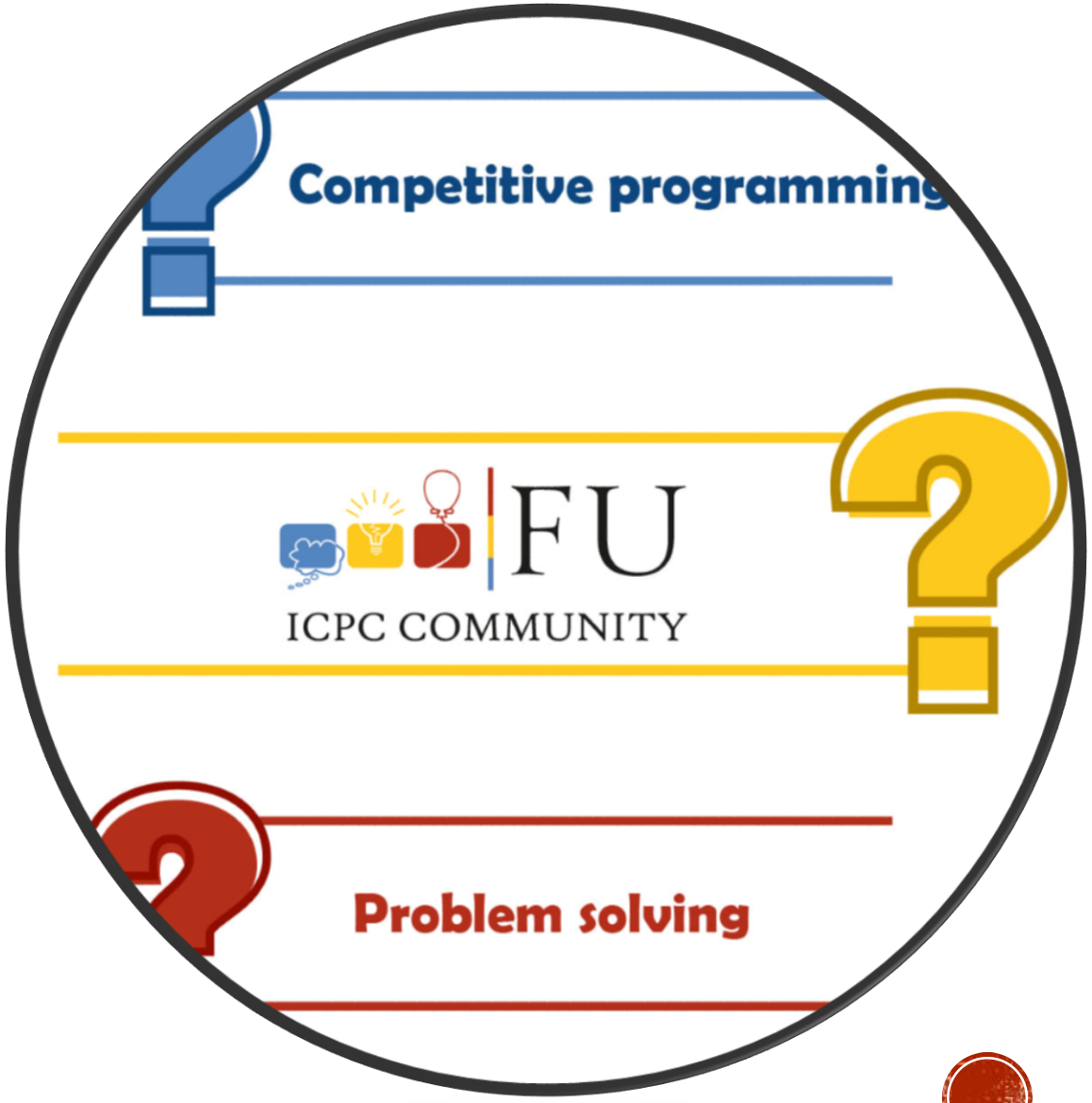
- Why are we here?
- compiler
- What is c++?
- Variables & Data types
- Arithmetic Operators
- Assignment Operators
- Pre ,Post increment & Decrement



WHY ARE WE HERE?

- Our vision is to help you through the journey till you can compete in ECPC
- There will be 1 session on site . We will be explaining new programming concepts
- Another online session , revision and solving some problems
- What do you need?!

Laptop & **passion**.

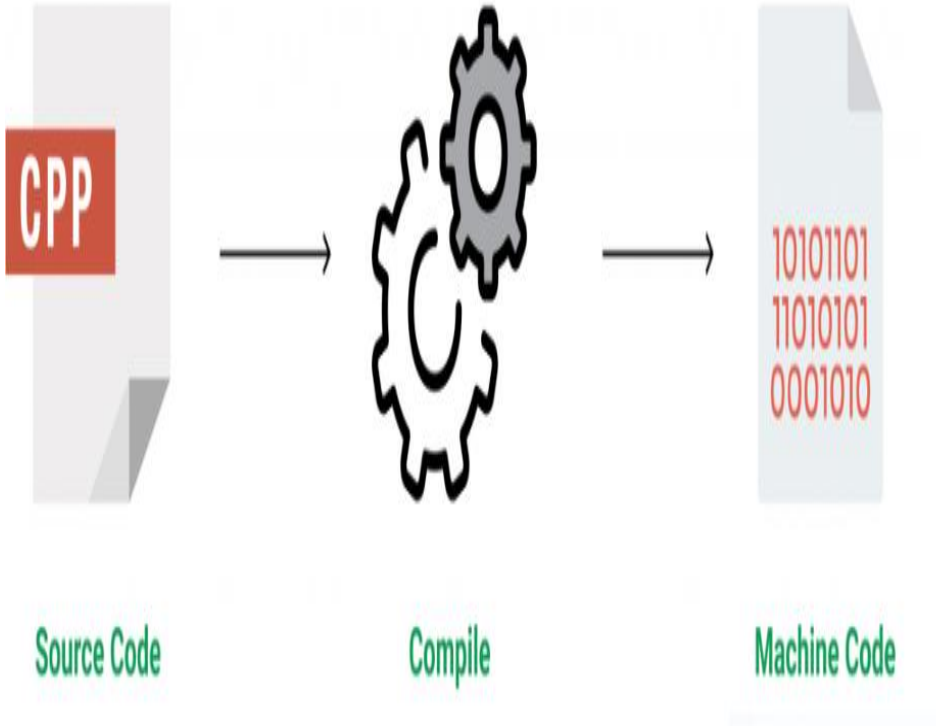


COMPILER

- A compiler is a special program that translates a programming language's source code into machine code.
- **Example**: C, C++, C#, Java.
- We will be using **C++ compiler**

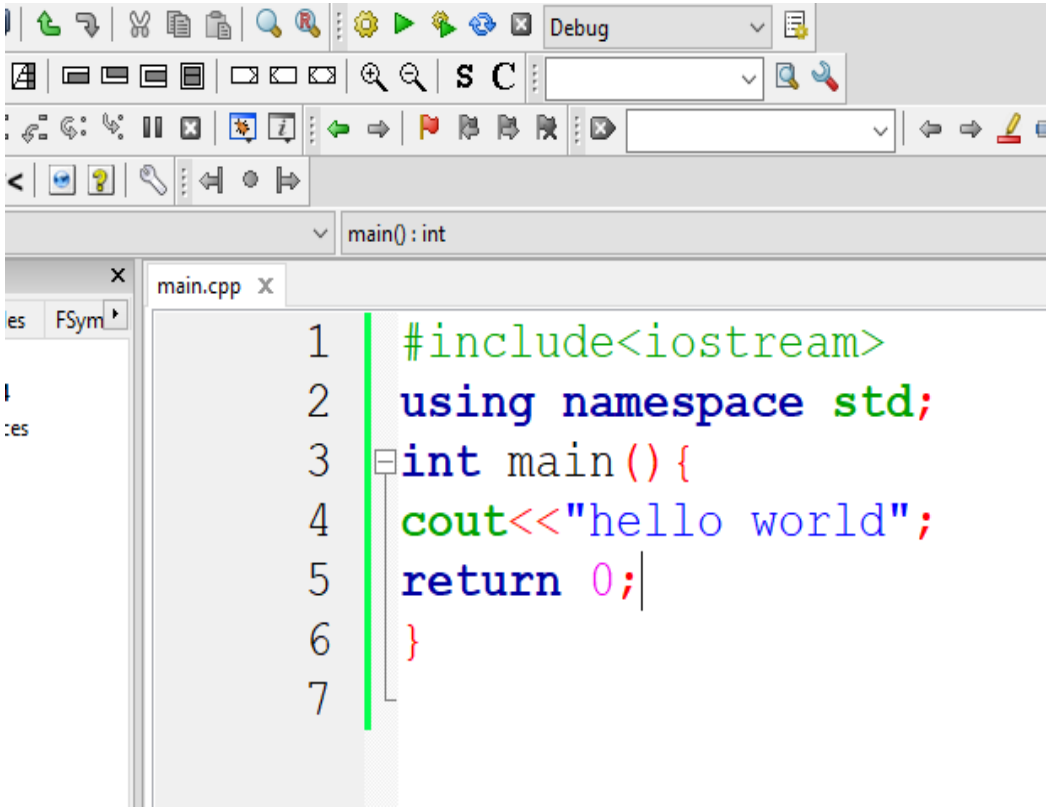
WHAT IS C++?

- C++ is a programming language that was enhancement of the C language to include object-oriented paradigm
- We understand human languages
- Computers understand machine language("Binary numbers")
- Programming language work as translators to the computer
- So we write in C++ and it translate it to the computer language then the computer do what we want.



WHAT IS C++?

- Has a rich library support (Both standard ~ built-in data structures, algorithms etc.)
- What is **#include**?!
- a way of including a standard or user-defined file in the program and is mostly written at the beginning of any C/C++ program.
- <iostream>?
- This is a standard library for input/output and will make me able to use some keywords to take input and print output



WHAT IS C++?

- **using namespace** std;???
- **namespace** is a collection of related names or identifiers (functions, class, variables) which helps to separate these identifiers from similar identifiers in other namespaces or the global namespace
- The identifiers of the C++ standard library are defined in a namespace called **std** **<u>like cout for printing</u>**
- int main(){}
- is the designated start of the program in hosted environment
- the known entry point when the run-time code is ready to start executing your program

main.cpp

```
#include<iostream>
int main(){

std::cout<<"Hello world";
return 0;
}
```

WHAT IS C++?

- In c++ at the end of each statement we put :

;

- cout<<“hello world”;
- cout<<“Samuel”;cout<<“Ramez”;
- cout<<“level 1”;
- **Comments in c++ :**
- // for single line
- For multiple line start with
- /*
- Hello fellow
- Welcome to ACM level 1
- */ end with

main.cpp

```cpp
#include <iostream>
int main() {
//this program is to print hello world
std::cout << "Hello World!";
/*
Hello fellows
welcome to ACM level 1
Good luck
*/
return 0;
}
```

WHAT IS C++?

- **Write a program to print “problem solving”**

WHAT IS C++?

- Write a program to print "problem solving"

```
#include<iostream>
using namespace std;
int main(){
cout<<"problem solving";
return 0;
}
```

```
"E:\sam's projects\contest4\bin\Debug\contest4.exe"
problem solving
Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.
```

VARIABLES & DATA TYPES

- Variables in C++ is a name given to a memory location. It is the basic unit of storage in a program.
- The value stored in a variable can be changed during program execution.
- A variable is only a name given to a memory location, all the operations done on the variable effects that memory location
- In C++, all the variables must be **declared** before use.
- Variables like boxes ! Carry what you put in there!

VARIABLES & DATA TYPES

- **How can we declare variables in C++?**

1. The name of the variable **contains** letters, digits, and underscores
2. All the variable names **must begin** with a letter of the alphabet or an underscore(_).
3. The name of the variable is **case sensitive** (ex Arr and arr both are different variables).
4. The name of the variable **does not contain** any whitespace and special characters (ex #,$,%,*, etc).
5. **We cannot use C++ keyword(ex float, double, class)as a variable name.**

VARIABLES & DATA TYPES

- **We cannot use C++ keyword(ex float, double, class)as a variable name.**

A – C	D – P	R – Z
alignas (C++11)	decltype (C++11)	reflexpr (reflection TS)
alignof (C++11)	default (1)	register (2)
and	delete (1)	reinterpret_cast
and_eq	do	requires (C++20)
asm	double	return
atomic_cancel (TM TS)	dynamic_cast	short
atomic_commit (TM TS)	else	signed
atomic_noexcept (TM TS)	enum	sizeof (1)
auto (1)	explicit	static
bitand	export (1) (3)	static_assert (C++11)
bitor	extern (1)	static_cast
bool	false	struct (1)
break	float	switch
case	for	synchronized (TM TS)
catch	friend	template
char	goto	this (4)
char8_t (C++20)	if	thread_local (C++11)
char16_t (C++11)	inline (1)	throw
char32_t (C++11)	int	true
class (1)	long	try
compl	mutable (1)	typedef
concept (C++20)	namespace	typeid
const	new	typename
consteval (C++20)	noexcept (C++11)	union
constexpr (C++11)	not	unsigned
constinit (C++20)	not_eq	using (1)
const_cast	nullptr (C++11)	virtual
continue	operator	void
co_await (C++20)	or	volatile
co_return (C++20)	or_eq	wchar_t
co_yield (C++20)	private	while
	protected	xor
	public	xor_eq

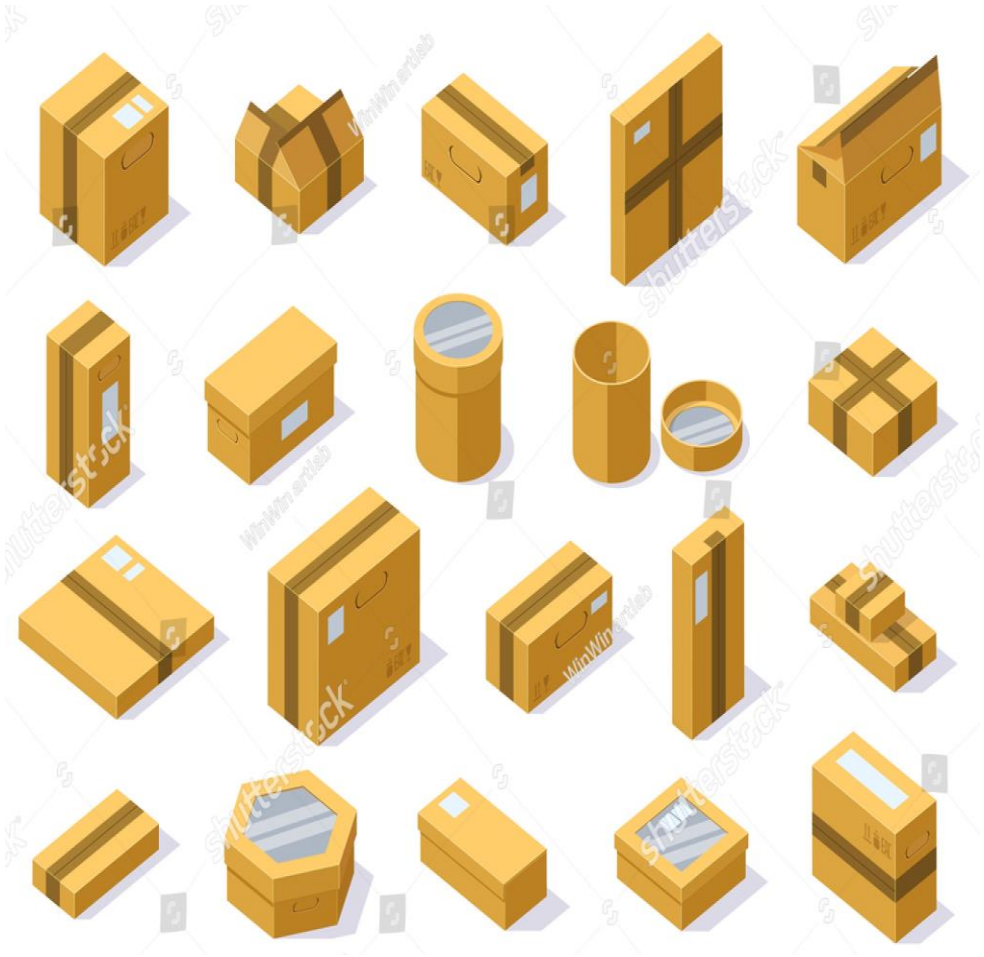
VARIABLES & DATA TYPES

- **How can we declare variables in C++?**
- sAm=
- Problem=
- 32p= ✘
- x=
- Xc12AF=
- _yY0=
- Y_123_uy=

What should we store in these boxes?

VARIABLES & DATA TYPES

- **Different type of boxes ”variables” carry different type of things !**
- What are different type of variables?
- Data types:
  - **Integer** 1,2,3,4 //int 4 bytes
  - **Long long** gives you more space//long long 8 bytes
  - **Float** 1.1,1.21,1.5,100.1 // float 4 bytes
  - **Double** 1.1 ,10^9.2//double 8bytes
  - **Bool true** or **false**// bool 1 byte
  - **String** “A”,”AC”,”ACM” //string size depend on the size of the string //24 byte
  - **Char** ‘A’ , ’a’,’+’,’?’ // char 1 byte

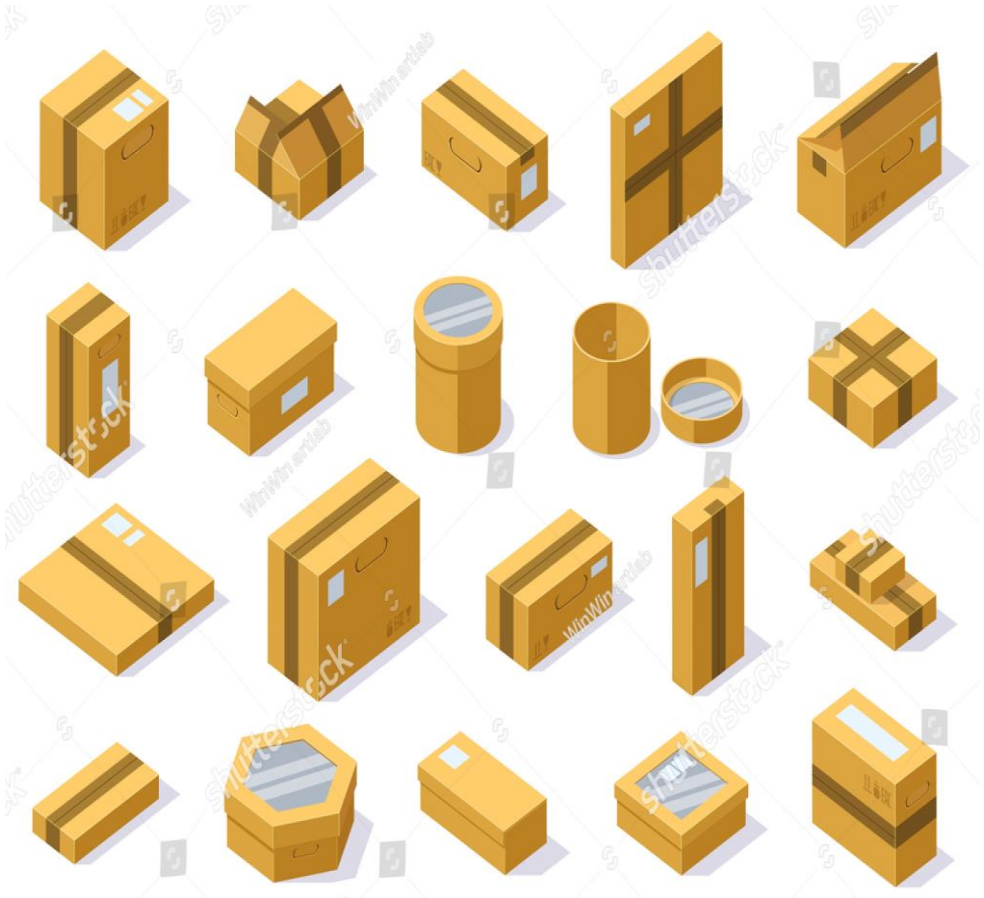


VARIABLES & DATA TYPES

- **How can we declare variables in C++?**
- int Sam=10;
- long long Y_123_uy=12345678999;
- float Problem=10.5;
- double num;
- num= 12345678999.5;
- 12p=//X
- string X="Samuel";
- string X="Ramez";
- bool X12=true;
- char _y='A';

Data type Variable_Name=value;

Data type Variable_Name;
Variable_Name =value;



VARIABLES & DATA TYPES

- **Write a program and declare 7 variables each variable with different data type then print all variables”**

VARIABLES & DATA TYPES

```
#include <iostream>
using namespace std;
int main()
{
    int x = 5;
    long long val2 = 310000093939;
    float val3 = 5.34;
    double val4 = 310009639.0124;
    char val5 = 'h';
    string val6 = "samuel";
    bool check = false;
    cout<<x;
    cout<<val2;
    cout<<val3;
    cout<<val4;
    cout<<val5;
    cout<<val6;
    cout<<check;
}
```

```
"E:\sam's projects\contest4\bin\Debug\contest4.exe"
5310000093939 5.343.1001e+08hsamuel0
Process returned 0 (0x0)   execution time : 0.036 s
Press any key to continue.
```

OPERATORS

Operator	Operation
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulo Operation (Remainder after division)

OPERATORS

- **Declare two variable, assign any values to them. Then apply each operator on them and print the result after each time**

OPERATORS

```
#include <iostream>
using namespace std;
int main() {
int a, b;
a = 7;
b = 2;
// printing the sum of a and b
 cout << "a + b = " << (a + b) << endl;
 // printing the difference of a and b
cout << "a - b = " << (a - b) << endl;
// printing the product of a and b
cout << "a * b = " << (a * b) << endl;
// printing the division of a by b
cout << "a / b = " << (a / b) << endl;
// printing the modulo of a by b
cout << "a % b = " << (a % b) << endl;
return 0; }
```

→

```
a + b = 9
a - b = 5
a * b = 14
a / b = 3
a % b = 1
```

OPERATORS

DIVISION OPERATOR /

- Division of different data types be like:
- int/int = int
- float /int =float
- int/float=float
- double /int =double
- int *long long=long long
- long long *double =double

```
In C++,

 7/2 is 3
7.0 / 2 is 3.5
7 / 2.0 is 3.5
7.0 / 2.0 is 3.5
```

OPERATORS

MODULO OPERATOR %

- The modulo operator % computes the remainder.
- 9%2=9-(9/2)*2=1
- Used to:
  - Last Digit .
  - Multiplication.
  - divisibility
  - primality.
- **Does Not Work on doubles**.

OPERATORS
ASSIGNMENT OPERATORS

Operator	Example	Equivalent to
`=`	`a = b;`	`a = b;`
`+=`	`a += b;`	`a = a + b;`
`-=`	`a -= b;`	`a = a - b;`
`*=`	`a *= b;`	`a = a * b;`
`/=`	`a /= b;`	`a = a / b;`
`%=`	`a %= b;`	`a = a % b;`

OPERATORS

ASSIGNMENT OPERATOR =

- // assign 5 to a variable and 6 to another one
- int a=5;
- int b=6;
- OR int a=5,b=5;
- a=b;
- b=18;
- cout<<a;??
- int x,y;
- x=y=a;
- cout<<x<<y<<a;??
- //assign ‘a’ to a variable
- char letter=‘a’;

OPERATORS

- **Declare two variable, assign any values to them . Then add both of them using assignment operator “+=”**

OPERATORS

```
#include <iostream>
using namespace std;
int main() {
 int a, b;
// 2 is assigned to a
a = 2;
// 7 is assigned to b
b = 7;
cout << "a = " << a << endl;
cout << "b = " << b << endl;
cout << "After a += b;" << endl;
// assigning the sum of a and b to a
 a += b;
// a = a +b
cout << "a = " << a << endl;
 return 0; }
```

→

```
a = 2
b = 7
After a += b;
a = 9
```

OPERATORS

INCREMANT & DECREMENT OPERATOR ++,--

- C++ also provides increment and decrement operators ++ and -- respectively.
- ++ increment the value by 1
- -- decrement the value by 1
- a++ is like a=a+1;
- a-- is like a=a-1;

OPERATORS

INCREMANT & DECREMENT OPERATOR ++,--

- int a=5;
- a++;//a=6
- --a;//a=5
- int b=6;
- ++b;//b=7
- b=18;
- b--;//b=17
- --a;//a=5

OPERATORS

PRE,POST- INCREMENT & PRE,POST-DECREMENT OPERATOR ++,--

- **Pre-increment operator**: operator used to increment the value of a variable **before using** it in an expression. In the Pre-Increment, value is first incremented and then used inside the expression.
- int x=5;
- a=++x;
- cout<<a;
- cout<<x;
- a=6
- x=6
- int b=7;
- int c=--b;
- cout<<c; cout<<b;
- c=6
- b=6

OPERATORS

PRE,POST- INCREMENT & PRE,POST-DECREMENT OPERATOR ++,--

- **Post-increment operator**: operator used to increment the value of the variable **after executing** the expression completely in which post-increment is used.
- int x=5;
- a=x++;
- cout<<a;
- cout<<x;
- a=5
- x=6
- int b=7;
- int c=b--;
- cout<<c; cout<<b;
- c=7
- b=6

REFERENCES

- First Project in C++
- Variables and Data types.
- Priorities&Calculations
- Basic Arithmetic&Casting
- Prefix and Postfix&Compound assignment
- Variable Scope (Local vs Global)

THANK YOU