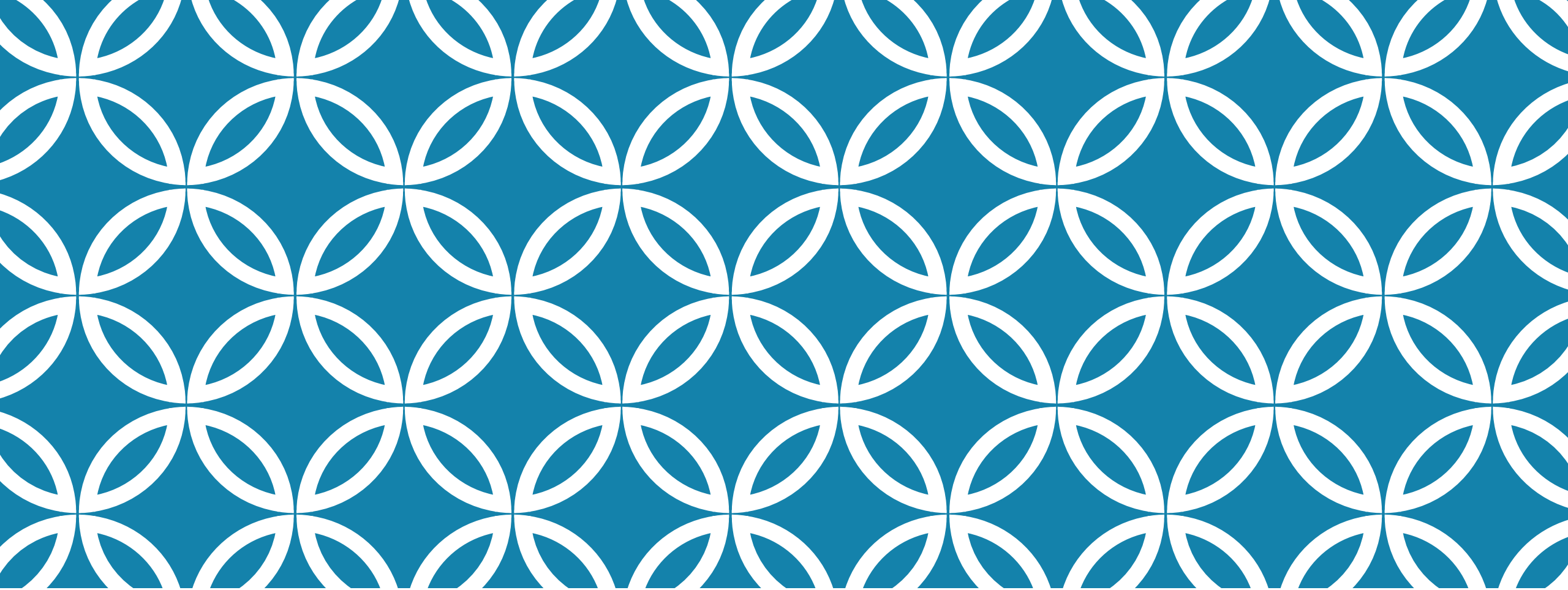


INTRODUCTION OF C++ SECTION 2

Dr/Ghada Maher
Eng/ Hossam Medhat

C++ COMMENTS

- Comments can be used to explain C++ code, and to make it more readable.
  It can also be used to prevent execution when testing alternative code.
- Comments can be singled-lined or multi-lined.

Example

```
// This is a comment
cout << "Hello World!";
```

C++ COMMENTS " CONT"

- **Multi-line comments start with /* and ends with */.**

Example

```
/* The code below will print the words Hello World!
to the screen, and it is amazing */
cout << "Hello World!";
```

C++ Variables

- Variables are containers for storing data values.
- Declaring (Creating) Variables :

Syntax

```
type variableName = value;
```

C++ Variables Examples

Example

Create a variable called **myNum** of type `int` and assign it the value **15**:

```
int myNum = 15;
cout << myNum;
```

Example

```
int myNum;
myNum = 15;
cout << myNum;
```

Example

```
int myNum = 15;  // myNum is 15
myNum = 10;  // Now myNum is 10
cout << myNum;  // Outputs 10
```

C++ Variables

- **Declare Many Variables :**
- **To declare more than one variable of the same type, use a comma-separated list:**

Example

```
int x = 5, y = 6, z = 50;
cout << x + y + z;
```

C++ IDENTIFIERS

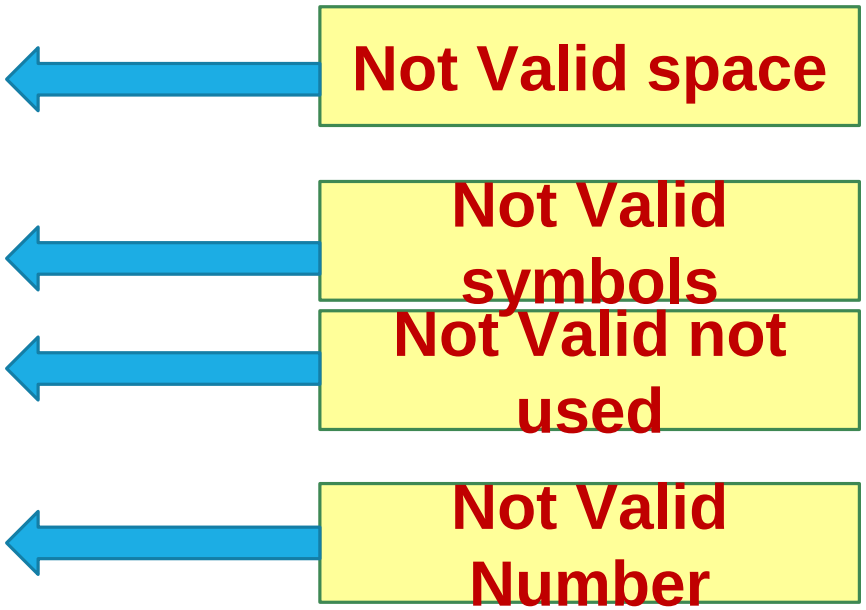
❖**The general rules for naming variables are:**

1) C++ variables must be identified with unique names.
2) Names can contain letters, digits and underscores.
3) Names must begin with a letter or an underscore (_) or dollar sign $
4) Names are case sensitive (myVar and myvar are different variables)
5) Names cannot contain whitespaces or special characters like !, #, %, etc.
6) Names cannot begin with number.
7) Reserved words (like C++ keywords, such as int) cannot be used as names

C++ IDENTIFIERS EXAMPLES

❖**Identify the valid and not valid variable names :**

1) Hossam
2) _H1
3) Group one
4) Prise$
5) (area)
6) int
7) Int_type
8) 2Teams



C++ CONSTANTS

- **Const keyword (this will declare the variable as "constant", which means unchangeable and read-only):**

Example

```
const int myNum = 15;   // myNum will always be 15
myNum = 10;   // error: assignment of read-only variable 'myNum'
```

Example

```
const int minutesPerHour = 60;
const float PI = 3.14;
```

C++ USER INPUT

- **cin is a predefined variable that reads data from the keyboard with the extraction operator (>>).**

Example

```
int x;
cout << "Type a number: "; // Type a number and press enter
cin >> x; // Get user input from the keyboard
cout << "Your number is: " << x; // Display the input value
```

```
Type a number:
```

```
Type a number: 2
Your number is: 2
```

C++ USER INPUT

Question1 : Tracing the code and Show what is result that output on

main.cpp

```cpp
#include <iostream>

using namespace std;

int main()
{
    int x , y ;
    cout << "x = " ;
    cin >> x ;
     cout << "y = " ;
    cin >> y;
    cout <<" The output is = "<< x/y << endl ;
    return 0;
}
```

When x = 10 , y = 4

The output = 2

Or

When x = 10.0 , y = 4.0

The output = ???

C++ USER INPUT

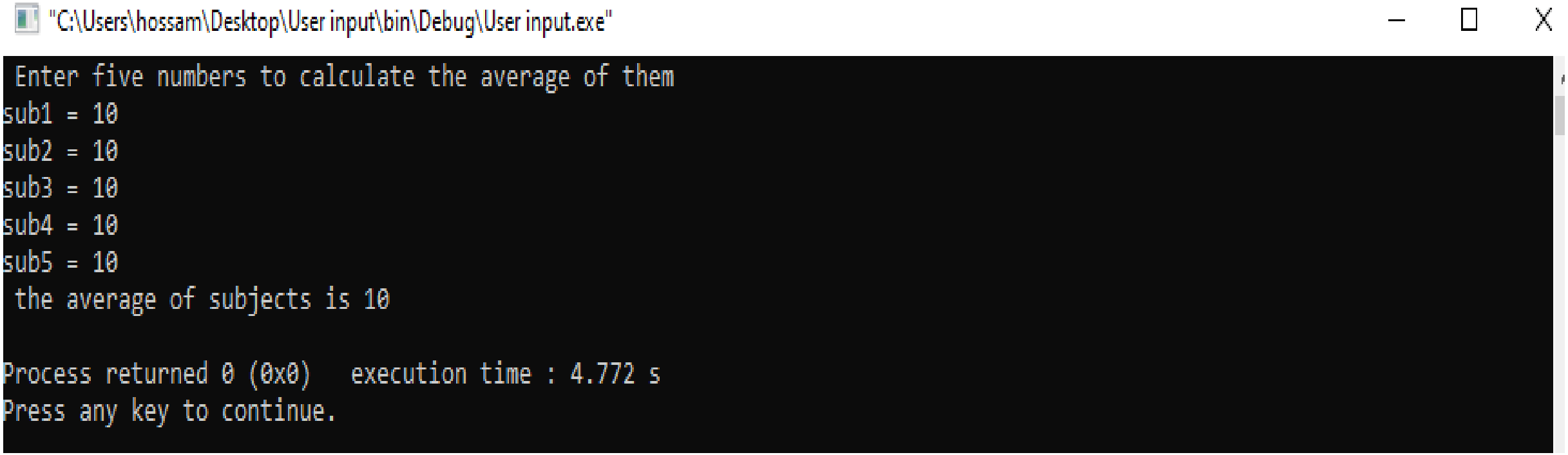
Question2 : Write The Program That Add Two Numbers with User

```cpp
#include <iostream>
using namespace std;

int main() {
  int x, y;
  int sum;
  cout << "Type a number: ";
  cin >> x;
  cout << "Type another number: ";
  cin >> y;
  sum = x + y;
  cout << "Sum is: " << sum;
  return 0;
}
```

```
Type a number: 5
Type another number: 5
Sum is: 10
```

C++ USER INPUT

Question3 : Write The Program That Calculate the average of five Numbers with User Input



C++ DATA TYPES

❖The data type specifies the size and type of information the variable will store:

1) Numeric Types :

- Use int or short when you need to store a whole number without decimals,

  like 35 or 1000.

- int (4 bytes) Vs short (2 bytees) .
- Use float or double when you need a floating point number (with decimals),

  like 9.99 or 3.14515.

- float (4 bytes) Vs double (8 bytees) .

C++ Data Types Examples

- **what is the output of this code ?**

Example

```
float f1 = 35e3;
double d1 = 12E4;
cout << f1;
cout << d1;
```

35000
120000

C++ DATA TYPES " CONT "

2) Boolean Types

- A boolean data type is declared with the bool keyword and can only take the values true or false.
- When the value is returned, true = 1 and false = 0.

```
#include <iostream>
using namespace std;

int main() {
  bool isCodingFun = true;
  bool isFishTasty = false;
  cout << isCodingFun << "\n";
  cout << isFishTasty;
  return 0;
}
```

```
1
0
```

C++ BOOLEAN EXPRESSIONS

- A Boolean expression is a C++ expression that returns a boolean value: 1 (true) or 0 (false).
- You can use a comparison operator, such as the greater than (>) operator to find out if an expression (or a variable) is true:

```
#include <iostream>
using namespace std;

int main() {
  int x = 10;
  int y = 9;
  cout << (x > y);
  return 0;
}
```

```
1
```

C++ BOOLEAN EXPRESSIONS

- A Boolean expression is a C++ expression that returns a boolean value: 1 (true) or 0 (false).
- You can use a comparison operator, such as the greater than (>) operator to find out if an expression (or a variable) is true:

```
#include <iostream>
using namespace std;

int main() {
  int x = 10;
  cout << (x == 10);
  return 0;
}
```

```
1
```

C++ BOOLEAN EXPRESSIONS

- A Boolean expression is a C++ expression that returns a boolean value: 1 (true) or 0 (false).
- You can use a comparison operator, such as the greater than (>) operator to find out if an expression (or a variable) is true:

```
#include <iostream>
using namespace std;

int main() {
  cout << (10 == 15);
  return 0;
}
```

```
0
```

C++ DATA TYPES " CONT "

3) Character Types

- The char data type is used to store a single character.
- The character must be surrounded by single quotes, like 'A' or

```
#include <iostream>
using namespace std;

int main () {
  char myGrade = 'B';
  cout << myGrade;
  return 0;
}
```

```
B
```

C++ DATA TYPES " CONT "

3) String Types

- The string type is used to store a sequence of characters (text).
- String values must be surrounded by double quotes (" "):.

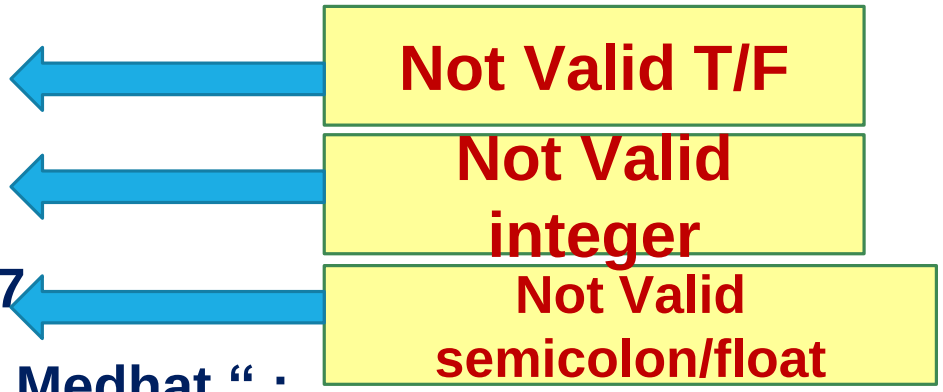
```
#include <iostream>
#include <string>
using namespace std;

int main() {
  string greeting = "Hello";
  cout << greeting;
  return 0;
}
```

```
Hello
```

C++ DATA TYPES EXAMPLES

❖**Which of these statement is wrong ?**

1) char me = ‘ I ’ ;
2) bool fact = True ;
3) bool numbers = 17;
4) int price = 23.5 ;
5) double number = 1.7
6) string x = “ hossam Medhat “ ;



C++ OPERATORS

- **Operators are used to perform operations on variables and values.**
- **C++ divides the operators into the following groups:**

1) **Arithmetic operators.**
2) **Assignment operators.**
3) **Comparison operators.**
4) **Logical operators.**

ARITHMETIC OPERATORS

- Arithmetic operators are used to perform common mathematical operations.

Operator	Name	Description	Example
+	Addition	Adds together two values	x + y
-	Subtraction	Subtracts one value from another	x - y
*	Multiplication	Multiplies two values	x * y
/	Division	Divides one value by another	x / y
%	Modulus	Returns the division remainder	x % y
++	Increment	Increases the value of a variable by 1	++x
--	Decrement	Decreases the value of a variable by 1	--x

ASSIGNMENT OPERATORS

- The addition assignment operator (+=) adds a value to a variable:

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
^=	x ^= 3	x = x ^ 3

COMPARISON OPERATORS

- Comparison operators are used to compare two values.
- The return value of a comparison is either true (1) or false (0).
- A list of all comparison operators:

Operator	Name	Example
==	Equal to	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

LOGICAL OPERATORS

- Comparison operators are used to compare two values.
- The return value of a comparison is either true (1) or false (0).
- A list of all comparison operators:

Operator	Name	Description	Example
&&	Logical and	Returns true if both statements are true	x < 5 && x < 10
||	Logical or	Returns true if one of the statements is true	x < 5 || x < 4
!	Logical not	Reverse the result, returns false if the result is true	!(x < 5 && x < 10)

QUESTION1 :

❖**Tracing the code and Show what is result that output on the screen ?**

```
#include <iostream>
using namespace std;

int main() {
   int x = 0 ;
   int y = 4 ;
   double z = 3.0 ;
   x+= 2 ;
   z*= x + 4 - 7 ;
   y+= x * 3 ;
   cout << "x = " << x << endl;
   cout << "y = " << y << endl;
   cout << "z = " << z << endl;
   return 0;
}
```

X=2
Y=10
Z=-3.0

QUESTION2 :

❖**Tracing the code and Show what is result that output on the screen ?**

```
#include <iostream>
using namespace std;

int main() {
  int x = 0 ;
  x++;
  --x;
  x+= 2 ;
  cout << x++ << endl;
  cout << --x << endl;
  return 0;
}
```

8

8

THANKS

Dr/Ghada Maher
Eng/ Hossam Medhat