Week 4

Ms. Kanwal lodhi

Types of main() function

- Every program has an entry and an exit point. An entry point is from where the program starts its execution, the exit point is where the program would terminate.
- Now in C++, that entry point from where the Operating system understands that this is the starting point to run and execute the code, is understood by the **main()** function.
- Hence, every C++ program has to have a main() function in it.
- Types : int main() , void main(), int main(void) are nothing but *function definitions* of **main()**.

void main()

- Syntax:
- #include<iostream.h>

  void main()

  {

  cout<<" Hello World!";

  }
- Output : Hello World!
- In the above syntax, '***void***' is the return type of the function. void means null in C/C++. Hence the function does not return any value to the Operating system after its execution, that is on exit.
- Here the only issue arises is to know if the program ran successfully or not.

int main()

- **Syntax:**
- #include<iostream.h>

```
int main()
{
 cout<<" Hello World!";
return 0;
 }
```

- **Output:** Hello World!
- Here, the return type of main is ***int.*** That is the function ideally expects a return type value int (integer) to be passed to it. Hence, the return statement; which returns value 0 to main on completion of the program.
- The purpose of returning a value **0** to main is for the operating system to understand that the program has executed **successfully**.

Basic INPUT/ OUTPUT

- The statement that are used to get data and then to assign it to variables are known as input statements.
- The statements that are used to receive data from the computer memory and then to send it to the output devices are called output statements.

Output Stream

- The “cout object” is C++ predefined object stands for “console output” where console means the computer display screen.
- It is part of iostream header file.
- Flow of data from one location to another is called stream. The cout is the standard output stream of C++.
- Syntax:

  cout<<const1/var1<<const2/var2…..;
- The string constants are given in double quotation marks. Numeric , variable and expressions are given without quotation marks.
- For each variable and constant separate << is used.
- Example: cout<<“one kilobyte=“<<1024<<“bytes”;

  Output : one kilobyte=1024bytes

Input Stream

- The “cin” stands for console input. It is input stream.
- It is used as input statement to get input from the keyboard during execution of the program.
- When a value is typed and Enter key is pressed, the value is assigned to the variable and control shifts to the next statement.
- The “cin” object is also a part of iostream header file.
- cin>>var1>>var2……;
- Each variable is separated by extraction operator “>>”.
- At least one variable on the right hand side of the “>>” operator must be used.
- example : cin>>a>>b>>c;
- This statement takes the values of a, b , c.

Assignment Statement

- The statement that is used to assign a value to variable is called the assignment statement.
- The assignment statement evaluates an expression and then assigns its value to a variable.
- The assignment operator “=“ is used to assign the calculated value to a variable.
- Syntax :

  var = expression;

  - var is the variable name to which the value of expression is assigned.
  - The variable type must be same as the value returned from it.
  - expression

Assignment Statement

Compound Assignment Statement

- The assignment statement can also be used to assign one value to many variables. This type of assignment statement is called compound assignment statement.
- Example : x=y=16;

Compound Assignment Expression

- The expression is use to add, subtract, multiply or divide a value to or from a variable without writing the variable on either side of the assignment operator '='.
- Syntax : var op = expression.
- For example :

  x = x+10; can be written in compound statement as

  x + = 10;

Operators

- Operators are used to perform operations on variables and values.
- **Operators in C++ can be classified into 6 types:**
  - Arithmetic Operators.
  - Relational Operators.
  - Logical Operators.
  - Bitwise Operators.
  - Assignment Operators.
  - Ternary or Conditional Operators.
- **Arithmetic Operators**:
  - The arithmetic operators are the symbols that represents arithmetic operations.
  - Each arithmetic operator operates upon two numeric values (constant or variables) and returns a value.

Arithmetic Operators

Operator	meaning	Examples
+	plus - Add two operands	x+y
-	Minus - subtract right operand from the left	x-y
*	Multiplication- multiply two operands	x*y
/	Division - devide left operand by the right one	x/y
%	Modulus - remainder of the division of left operand by the right	x%y

Arithmetic Operators and Expressions

- All arithmetic operators, except the remainder operator are used for all type of data.
- The remainder or modulus operator can only be used for integer type data. It returns the remainder when one integer is divided by another integer. For example : 7%2 returns remainder 1

- **Arithmetic Expressions**
  - **An** arithmetic expression is a combination of variables , constants and arithmetic operators.
  - it is used to calculate the value of an arithmetic formula and returns a single value.
  - Receiving variable: the variable on left side that receive the value of expression.
  - Assignment operator '=' is used in expression to assign result to variable.
  - For example c = a+b*3; here c is the receiving variable.

Order of Precedence of Operation

- If there are multiple operators in a single expression, the operations are not evaluated simultaneously. Rather, operators with higher **precedence** have their operations evaluated first.
- In C++ operation are performed in following order:
  - All multiplication , divisions and modulus are performed first from left to right.
  - All addition and subtraction are then performed from left to right.
  - If parenthesis are used in an expression , the expression within the parenthesis are first computed from left to right.
  - When parenthesis are used within parenthesis, the expression within inner most