Lecture 7: Programming Using C++

Write program:

Welcome
To
C++

1. Write a program to write the text given above using 3 cout statement.
2. Write a program to write the text given above using 1 cout statement.

Variables:

- Variables are like containers in your computer’s memory - you can store values in them and retrieve or modify them when necessary.
- To INITIALIZE a variable means to store a value in it for the first time, which is done using the ASSIGNMENT OPERATOR, like this: x = 2

Assignment:

- Putting a value to a variable.

number = 25;
sum = 23 + 56;
number = number + 1;

Naming Constants and Variables:

<table>
<thead>
<tr>
<th>Names...</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>CANNOT start with a number</td>
<td>2i</td>
</tr>
<tr>
<td>CAN contain a number elsewhere</td>
<td>h2o</td>
</tr>
<tr>
<td>CANNOT contain any arithmetic operators...</td>
<td>r*s+t</td>
</tr>
</tbody>
</table>
An Introduction to the 4 Data Types

- In C++, there are four basic DATA TYPES:
  - int
  - char
  - float
  - double

- Each one has its own properties. For instance, they all have different sizes.
- We must give each variable a data type to allow and restrict the type of data we can assign to it.

**cout:**

- Printing constant and variables:

  ...
  int number;
  number = 5;
  cout<"Number is "< number;
  ...

  Argument 1: “Number is ”, format control string
  Argument 2: number, the value to be printed.
cout:

**What is the output of the following statement?**

```cpp
int number1, number2;
number1=5;
number2=8;
cout<<"number 1 =" <<number1<<"number2 ="<< number2;
```

**The int Data Type:**

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

main() {
    int a,b,c,d,e;
    a = 10;
    b = 4.3;
    c = 4.8;
    d='A';
e=4.3+4.8;
    cout<<"a="<<a<<"\n";
    cout<<"b="<<b<<"\n";
    cout<<"c="<<c<<"\n";
    cout<<"d="<<d<<"\n";
    cout<<"e="<<e<<"\n";
    cout<<"b+c="<<b+c<<"\n";
}
```

**The output of the example is:**

- a=10
- b=4
- c=4
- d=65
- e=9
- b+c = 8
cin:

- It is used to get a value from user (keyboard).

```cpp
cin>>number1;
cout<<“Please enter a number: “;
cin>>number1;
```

**Practice:**

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

int main()
{
    int number1, number2, sum;
    cout<<"Enter first integer : ";
cin>>number1;
    cout<<"Enter second integer : ";
cin>>number2;
    sum = number1 + number2;
    cout<<“Sum = ”<< sum<<“\n”;
}
```

**Programming guidelines:**

- Place a space after each comma to make programs more readable.
- Use meaningful variable and constant names (total, average, sum, etc.)
- Combine multiple-word variables like “total_commission” or “totalCommision”
- Start with a lowercase letter to a variable name.
- Do not forget that C++ is a case sensitive language.
- Do not place variable declarations among executable statements.
- Separate the declarations and executable statements with a blank line.
- Place spaces on either side of an operator.
sum = number1 + number2;
sum = number1 + number2; ← better

**Arithmetic operations in C++:**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Operator</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addition</td>
<td>+</td>
<td>a + b, 45 + 7</td>
</tr>
<tr>
<td>Subtraction</td>
<td>-</td>
<td>a − b, 45 − 7</td>
</tr>
<tr>
<td>Multiplication</td>
<td>*</td>
<td>a * b, 45 * 7</td>
</tr>
<tr>
<td>Division</td>
<td>/</td>
<td>a / b, 45 / 7</td>
</tr>
<tr>
<td>Modulus/remainder</td>
<td>%</td>
<td>a % b, 45 % 7</td>
</tr>
</tbody>
</table>

- **Integer division**

  ```
  int result;
  result = 17 / 5;    // result = 3
  result = 7 / 4;     // result = 1
  ```

- **Modulus /remainder**

  ```
  result = 17 % 5;    // result = 2
  result = 7 % 4;     // result = 3
  ```

**Operation** | **Operator**
--- | ---
Parentheses | ()
Multiplication, division remainder | *, /, %
Addition, subtraction | +, -