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>
<tr>
<td>Cannot contain any other punctuation marks...</td>
<td>@x%£!!a</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Can contain or begin with an underscore</td>
<td><em>height</em></td>
</tr>
<tr>
<td>Cannot be a C keyword</td>
<td>struct</td>
</tr>
<tr>
<td>Cannot contain a space</td>
<td>im stupid</td>
</tr>
<tr>
<td>Can be of mixed cases</td>
<td>XSquared</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:

```cpp
... 
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;

int main() {
    int a,b,c,d,e;
    a = 10;
    b = 4.3;
    c = 4.8;
    d='A';
    e=4.3+4.8;
    cout<<"a=\n";  
    cout<<"b=\n";
    cout<<"c=\n";
    cout<<"d=\n";
    cout<<"e=\n";
    cout<<b+c="\n";
    return 0;
}
```

**Note:** Remark that return statement is an important part of your program but your code may run without it as your compiler may restore it automatically.
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).

cin>>number1;

- Try to put an informative line before the input; this may increase user friendliness of your program.

cout<<"Please enter a number: ";
cin>>number1;

Practice:

/* Addition program */
#include <iostream>
using namespace std;
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:

- Use indentation.

```cpp
main()
{
    cout<<"welcome";
    cout<<"to C++! \n";
}
```

- Put an explanation at the beginning of each function.

```cpp
/* First program in C++ */
main()
{
    cout<<"welcome";
    cout<<"to C++! \n";
}
```

- 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.

```cpp
sum=number1+number2;
sum = number1 + number2; ← better
```
Arithmetic operations in C++:

- **Addition** + \( a + b \), 45 + 7
- **Subtraction** - \( a - b \), 45 - 7
- **Multiplication** * \( a \times b \), 45 * 7
- **Division** / \( a / b \), 45 / 7
- **Modulus/remainder** % \( a \% b \), 45 % 7

- **Integer division**

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

- **Modulus /remainder**

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

<table>
<thead>
<tr>
<th>Operation</th>
<th>Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parentheses</td>
<td>()</td>
</tr>
<tr>
<td>Multiplication, division</td>
<td>* , / , %</td>
</tr>
<tr>
<td>remainder</td>
<td></td>
</tr>
<tr>
<td>Addition, subtraction</td>
<td>+ , *</td>
</tr>
</tbody>
</table>

Order of evaluation (Precedence):

- **Evaluate the order of the following operations:**

\[
a \times (b + c) + c \times (d + e)
\]
\[
y = a \times x \times b + c \times d + e
\]
\[
g = a + b / c + d \% e - (f + g) + a \times b / c
\]