Introduction to Computing

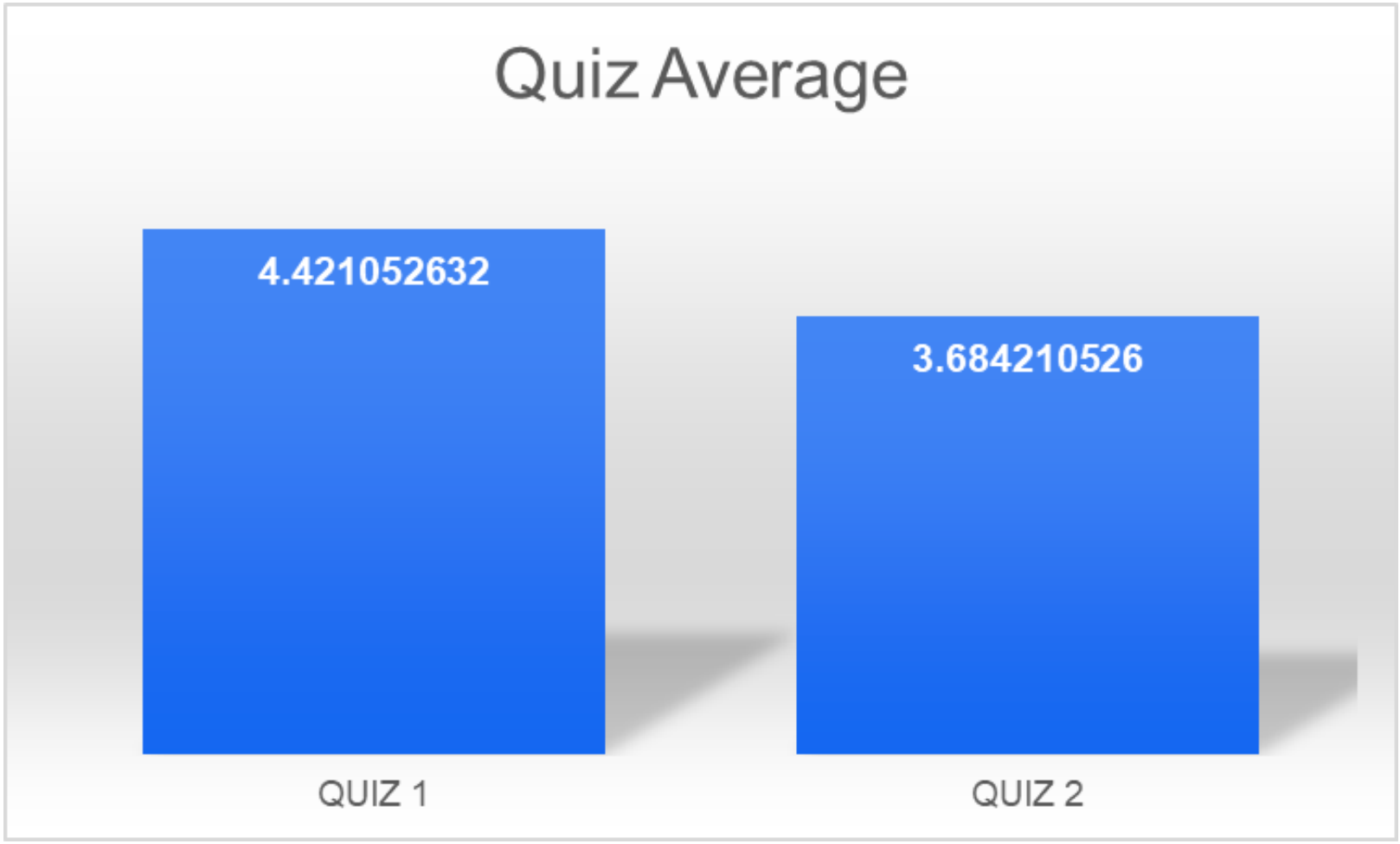
Lecture 5

Dr. Naveed Anwar Bhatti

Webpage: naveedanwarbhatti.github.io

C++ Operators

Administrivia



C++ Operators

- Operators are used to perform operations on **variables** and **values**.

- C++ divides the operators into the following groups:

  - **Arithmetic operators**
  - **Assignment operators**
  - **Comparison operators**
  - **Logical operators**
  - **Bitwise operators**

C++ 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 from 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

C++ Arithmetic Operators

Example:

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

int main()
{
    int a = 9, b = 4, c;

    c = a + b;
    cout << c << "\n";

    c = a - b;
    cout << c << "\n";

    c = a * b;
    cout << c << "\n";

    c = a / b;
    cout << c << "\n";

    c = a % b;
    cout << c << "\n";

    return 0;
}
```

C++ Arithmetic Operators

Example:

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

int main()
{
    int a = 10, b = 5;

    ++a;
    cout << a <<"\n";

    --b;
    cout << b << "\n";

    return 0;
}
```

C++ Assignment Operators

- Assignment operators are used to assign values to variables

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

C++ Assignment Operators

Example:

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

int main()
{
    int a = 5, c;
    c = a;       // c is 5
    cout << c << "\n";
    c += a;      // c is 10
    cout << c << "\n";
    c -= a;      // c is 5
    cout << c << "\n";
    c *= a;      // c is 25
    cout << c << "\n";
    c /= a;      // c is 5
    cout << c << "\n";
    c %= a;      // c = 0
    cout << c << "\n";
    return 0;
}
```

C++ Comparison Operators (Relational Operator)

- Comparison operators are used to compare two values
- The return value is either true (1) or false (0)

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

C++ Comparison Operators (Relational Operator)

Example:

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

int main()
{
    int a = 5, b = 5, c = 10;

    cout << (a == b) << "\n";
    cout << (a == c) << "\n";
    cout << (a > b) << "\n";
    cout << (a > c) << "\n";
    cout << (a < b) << "\n";
    cout << (a < c) << "\n";
    cout << (a != b) << "\n";
    cout << (a != c) << "\n";
    cout << (a >= b) << "\n";
    cout << (a >= c) << "\n";
    cout << (a <= b) << "\n";
    cout << (a <= c) << "\n";
    return 0;
}
```

C++ Logical Operators

- Logical operators are used to determine the logic between variables or values

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)

C++ Logical Operators

Example:

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

int main()
{
    int a = 5, b = 5, c = 10, result;

    result = (a == b) && (c > b);
    cout << result << "\n";
    result = (a == b) && (c < b);
    cout << result << "\n";
    result = (a == b) || (c < b);
    cout << result << "\n";
    result = (a != b) || (c < b);
    cout << result << "\n";
    result = !(a != b);
    cout << result << "\n";
    result = !(a == b);
    cout << result << "\n";
    return 0;
}
```

C++ Exercises

Question 1: Write a program in C++ to add two numbers print output

Sample Output:
59

Question 2: Write a program in C++ to add two numbers accept through keyboard

Sample Output:
Please enter first number: <29>
Please enter second number: <30>

The sum of 29 and 30 is : 59

Solution 1:

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

int main()
{
   int a;
   int b;
   int sum;
   a = 29;
   b = 30;
   sum = a + b;
   cout << sum << "\n";
}
```

Solution 2:

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

int main()
{
    int a, b, sum;

    cout << " Please enter first number: ";
    cin >> a;
    cout << " Please enter second number: ";
    cin >> b;
    sum = a + b;
    cout << " The sum of "<<a<<" and "<<b<<“ is : " << sum << endl;
    cout << endl;
    return 0;
}
```

Thanks a lot

If you are taking a Nap, **wake up**.....Lecture Over