Data Types

CS1004 OBJECT ORIENTED PROGRAMMING

Print

Comments

Running a program

Variables

Assignment

Rules for naming variables

Multiple assignments in a single line

```
#include<iostream>
Using namespace std;
int main()
{
  ◦ int age=25;
  ◦ float pi = 3.1415692;
  ◦ double salary = 5000.12;
  ◦ char ch = 'A';
}
```

Variables

Data Type	Size (bytes)
char	1
short	2
int	4
long	4
long long	8
float	4
double	8
bool	1
wchar_t	2

Variable Data size

```
#include<iostream>
Using namespace std;
int main()
{
  ◦ int age=25;
  ◦ age=50;
  ◦ age=age+5;
}
```

Assignment

Start with a letter or underscore: variable_name, myVariable

Contain only letters, digits, and underscores: valid_variable, number123

Case-sensitive: myVariable and myvariable are different variables

Avoid using reserved keywords: Do not use keywords like int, float, if, else, etc. as variable names.

Variable Name

Multiple assignment in single line

```
#include<iostream>

Using namespace std;

int main()

{
    int a,b,c;
    a=10;
    b=a+5+8+2;
    c=2*a+b;

}
```

Data types

Memory Limit

Possible Values

sizeof(x)

Numeric Data Types (int, float)

Integers (int,long)

Floating-point numbers (float,double)

Basic arithmetic operations on numeric types

Converting between numeric types

Text Data Type (str)

Characters

Strings (str)

Creating and manipulating strings

String concatenation and formatting

Common string methods (length(), etc.)

Basic Input/Output Operations

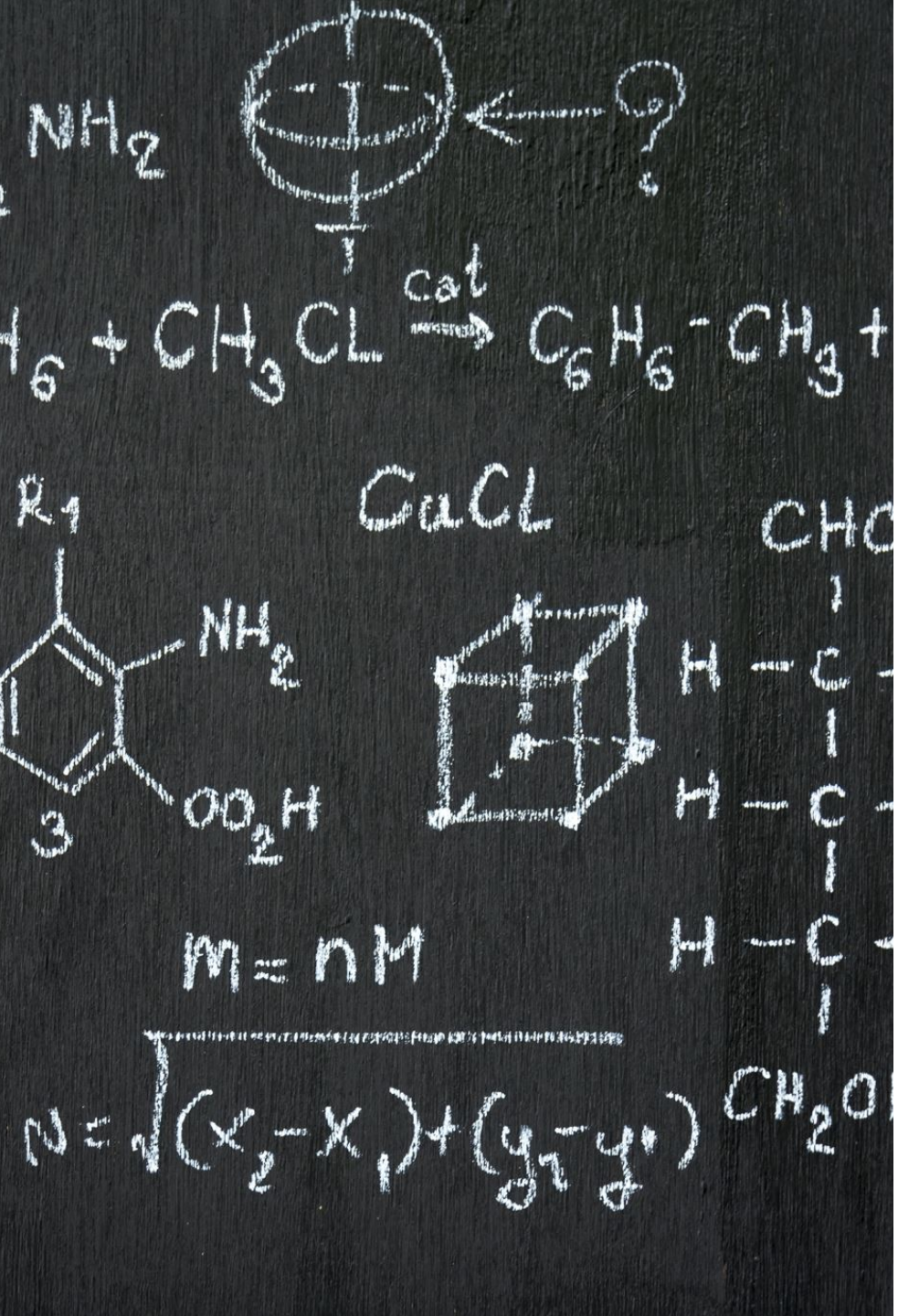
Using the cout function for output

Getting user input with cin

Formatting output strings

Boolean Data Type (bool)

- Booleans (bool)
- Understanding True and False values
- Boolean operations (and, or, not)
- Comparison operators in C++
- Practical uses of boolean data types



Arithmetic Operations

```
int sum = a + b;

int remainder = a % b;

int x=5;

x++;

y=x++; //(y=5,x=6)

x--;

++x;

--x;
```

Conditional Operators

IF ELSE AND SWITCH CASES

Conditional Operations if-else

```
int num = 10;
if (num > 0) {
    cout << "Number is positive" << endl;
} else {
    cout << "Number is negative or zero" << endl;
}
```

Nested if else

```
int age = 25;
if (age >= 18) {
    if (age >= 65) {
        cout << "You are a senior citizen." << endl;
    } else {
        cout << "You are an adult." << endl;
    }
} else {
    cout << "You are a minor." << endl;
}
```

if-else-if

```
int grade = 90;
if (grade >= 90) {
    cout << "Grade: A" << endl;
} else if (grade >= 80) {
    cout << "Grade: B" << endl;
} else if (grade >= 70) {
    cout << "Grade: C" << endl;
} else {
    cout << "Grade: F" << endl;
}
```

Conditional operator (ternary operator)

```
int x = 10;
```

```
int y = (x > 0) ? 1 : 0;
```

```
cout << "y: " << y << endl;
```

Switch Cases

```
int choice = 2;

   switch (choice) {

      case 1:

         cout << "You chose option 1" << endl;

         break;

      case 2:

         cout << "You chose option 2" << endl;

         break;

      default:

         cout << "Invalid choice" << endl;

   }
```

Multi cases

```
switch (choice) {
    case 1:
    case 2:
        cout << "You chose option 1 or 2" << endl;
        break;
    case 3:
        cout << "You chose option 3" << endl;
        break;
    default:
        cout << "Invalid choice" << endl;
}
```

Switch case with ranges

```
switch (choice) {

    case 1 ... 3:

        cout << "You chose a number between 1
and 3" << endl;

        break;

    default:

        cout << "Invalid choice" << endl;

}
```

Switch case with expressions

```
switch (x % 2) {
    case 0:
        cout << "x is even" << endl;
        break;
    case 1:
        cout << "x is odd" << endl;
        break;
}
```

Switch case with nested cases

```
switch (choice) {
    case 1:
        cout << "You chose option 1" << endl;
        break;
    case 2:
        cout << "You chose option 2" << endl;
        switch (anotherChoice) {
            case 1:
                cout << "Sub-option 2.1" << endl;
                break;
            case 2:
                cout << "Sub-option 2.2" << endl;
                break;
            default:
                cout << "Invalid sub-choice" << endl;
        }
        break;
    default:
        cout << "Invalid choice" << endl;
}
```

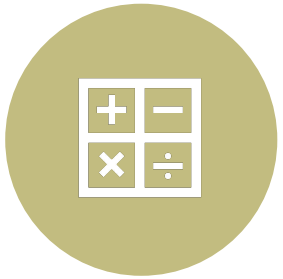
Arrays and Lists

Arrays

int numbers[5] = {1, 2, 3, 4, 5}; // Array of integers

int firstElement = numbers[0]; // Access the first element

int lastElement = numbers[4]; // Access the last element



numbers[2] = 10; // Change the third element to 10

Vector

```
#include <vector>

vector<int> myList = {1, 2, 3, 4, 5};
int firstElement = myList[0];
int lastElement = myList.back();

myList[2] = 10;
myList.push_back(6); // Add an element to the end
```

Loops

FOR WHILE AND DO WHILE

Loops

```
int numbers[5] = {1, 2, 3, 4, 5};

for (int i = 0; i < 5; i++) {
    cout << numbers[i] << " ";
}
```

```
int numbers[5] = {1, 2, 3, 4, 5};
int i = 0;

while (i < 5) {
    cout << numbers[i] << " ";
    i++;
}
```

Loops

```
int numbers[5] = {1, 2, 3, 4, 5};
int i = 0;

do {
    cout << numbers[i] << " ";
    i++;
} while (i < 5);
```

```
int matrix[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};


for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
        cout << matrix[i][j] << " ";
    }
    cout << endl;
}
```

Loops with if else

Print series (arithmetic, geometric, etc)

Sum of series (arithmetic, geometric)

Array processing with loops

Factorial using loops