Programming Fundamentals with C++

Detailed notes and example code for beginner programmers learning DSA fundamentals. Each topic includes explanations and complete C++ code.

1. Variables and Data Types

Variables store data and have specific data types.

C++ supports several basic data types:

- int: for integer numbers, e.g., 5, -20
- float: for floating-point numbers with decimals, e.g., 3.14
- double: double precision floating point
- char: stores a single character, e.g., 'A'
- bool: stores boolean values true or false

You must declare the type before using a variable.

Example shows how to declare and use variables:

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

int main() {
    int age = 25;
    float price = 99.99f;
    double pi = 3.14159;
    char grade = 'A';
    bool isPassed = true;

    cout << "Age: " << age << endl;
    cout << "Price: " << price << endl;
    cout << "Pi: " << pi << endl;
    cout << "Grade: " << grade << endl;
    cout << "Passed: " << isPassed << endl;
```

```
    return 0;
}
```

2. If-else / Switch Statements

Conditional statements allow decision making.

If-else checks conditions sequentially.

Switch works with discrete values like characters or integers.

Example of if-else checking if a number is positive, zero or negative:

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

int main() {
    int num;
    cout << "Enter a number: ";
    cin >> num;

    if (num > 0) {
        cout << "Positive number" << endl;
    } else if (num == 0) {
        cout << "Zero" << endl;
    } else {
        cout << "Negative number" << endl;
    }

    return 0;
}
```

Example of switch statement using grades:

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

int main() {
    char grade;
    cout << "Enter grade (A-F): ";
```

```
    cin >> grade;

    switch (grade) {
        case 'A':
        case 'a':
            cout << "Excellent!" << endl;
            break;
        case 'B':
        case 'b':
            cout << "Good job" << endl;
            break;
        case 'C':
        case 'c':
            cout << "You passed" << endl;
            break;
        case 'F':
        case 'f':
            cout << "Better luck next time" << endl;
            break;
        default:
            cout << "Invalid grade" << endl;
    }

    return 0;
}
```

3. Loops (for, while)

Loops repeat blocks of code multiple times.

For loops are used when number of iterations is known.

While loops are used when you repeat until a condition changes.

Example of a for loop printing numbers 1 to 5:

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

int main() {
    for (int i = 1; i <= 5; i++) {
        cout << i << " ";
```

```
    }
    cout << endl;

    return 0;
}
```

Example of while loop printing numbers 1 to 5:

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

int main() {
    int i = 1;
    while (i <= 5) {
        cout << i << " ";
        i++;
    }
    cout << endl;

    return 0;
}
```

4. Functions

Functions are reusable blocks of code that perform a specific task.

They help in organizing and structuring programs.

Functions can take parameters and return values.

Example: Function to add two numbers and return the sum:

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

// Function to add two numbers
int add(int a, int b) {
    return a + b;
}
```

```
int main() {
    int x = 5, y = 10;
    int result = add(x, y);
    cout << "Sum is: " << result << endl;

    return 0;
}
```

5. OOP Basics (class, object, inheritance)

Object-Oriented Programming (OOP) models real-world entities as classes and objects.

A class defines a blueprint for objects.

Objects are instances of classes.

Inheritance allows a class to inherit properties and behavior from another class.

Example of a base class Animal and derived class Dog overriding a method:

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

class Animal {
public:
    void sound() {
        cout << "Animal makes a sound" << endl;
    }
};

class Dog : public Animal {
public:
    void sound() {
        cout << "Dog barks" << endl;
    }
};

int main() {
    Animal a;
    Dog d;
```

```
    a.sound();  // Output: Animal makes a sound
    d.sound();  // Output: Dog barks

    return 0;
}
```