12.	Multiple Inheritance	It supports only single inheritance. Multiple inheritances are achieved partially using interfaces.	It supports both single and multiple Inheritance.
13.	Overloading	It supports only method overloading and doesn't allow operator overloading.	It supports both method and operator overloading.
14.	Pointers	It has limited support for pointers.	It strongly supports pointers.
15.	Libraries	It doesn't support direct native library calls but only Java Native Interfaces.	It supports direct system library calls, making it suitable for system-level programming.
16.	Libraries	Libraries have a wide range of classes for various high-level services.	C++ libraries have comparatively low-level functionalities.
17.	Documentation Comment	It supports documentation comments (e.g., /**.. */) for source code.	It doesn't support documentation comments for source code.
18.	Thread Support	Java provides built-in support for multithreading.	C++ doesn't have built-in support for threads, depends on third-party threading libraries.
19.	Type	Java is only an object-oriented programming language.	C++ is both a procedural and an object-oriented programming language.
20.	Input-Output mechanism	Java uses the (System class): **System.in** for input and **System.out** for output.	C++ uses **cin** for input and **cout** for an output operation.

22.	Structures and Unions	Java doesn't support Structures and Unions.	C++ supports Structures and Unions.
23.	Parameter Passing	Java supports only the Pass by Value technique.	C++ supports both Pass by Value and pass by reference.
24.	Inheritance Tree	All classes in Java are subclasses of the Object class, hence Java only ever follows a single inheritance tree.	A fresh inheritance tree is always created in C++.
25.	Global Scope	It supports no global scope.	It supports both global scope and namespace scope.
26.	Object Management	Automatic object management with garbage collection.	It supports manual object management using new and delete.
27.	Call by Value and Call by reference	Java supports only call by value.	C++ both supports call by value and call by reference.
28.	Hardware	Java is not so interactive with hardware.	C++ is nearer to hardware.
29.	Language Used for	Internet and Android games, Mobile applications, Healthcare and research computation, Cloud applications, Internet of Things (IoT) devices, etc.	Game engines, Machine learning, Operating systems, Google Search Engine, Web browsers, Virtual Reality (VR), Game development, Medical technology, Telecommunications, Databases, etc.

```
        return 0;
}
```

This will display the result of **student1**. The complete program is shown in Program 8.3.

Program 8.3 Multilevel Inheritance

```
#include <iostream>

using namespace std;

class student
{
    protected:
        int roll_number;
    public:
        void get_number(int);
        void put_number(void);
};

void student :: get_number(int a)
{
        roll_number = a;
}
void student :: put_number()
{
        cout << "Roll Number: " << roll_number << "\n";
}
class test : public student          // First level derivation
{
    protected:
        float sub1;
```

(Contd.)

```
}
class resu
{
    floa
  publi
    voi
};
void resu
{
}
int main
{
}
```

The output

```
Roll N
Marks
Marks
Total
```

```
    float sub2;
  public:
    void get_marks(float, float);
    void put_marks(void);
};

void test :: get_marks(float x, float y)
{
    sub1 = x;
    sub2 = y;
}

void test :: put_marks()
{
    cout << "Marks in SUB1 = " << sub1 << "\n";
    cout << "Marks in SUB2 = " << sub2 << "\n";
}

class result : public test          // Second level derivation
{
    float total;                    // private by default
  public:
    void display(void);
};

void result :: display(void)
{
    total = sub1 + sub2;
    put_number();
    put_marks();
    cout << "Total = " << total << "\n";
}
int main()
{
    result student1;                // student1 created

    student1.get_number(111);

    student1.get_marks(75.0, 59.5);

    student1.display();

    return 0;
}
```

The output of Program 8.3 would be:

```
Roll Number: 111
Marks in SUB1 = 75
Marks in SUB2 = 59.5
Total = 134.5
```