

## Stream Operators

```
/*
    . Stream insertion: std::cout << p ;
    . Stream extraction: std::cin >> p ;
    . Custom formatters for fmt

*/
```

## Stream insertion and extraction

```
// Stream insertion
Point p1(10, 20);
Point p2(3, 4);

std::cout << p1 << "\n";
std::cout << p2 << "\n";
```

```
// Stream extraction
Point p2;
std::cin >> p2; //
std::cout << p2 << "\n";
```

## Stream insertion and extraction

```
export class Point
{
    friend std::ostream &operator<<(std::ostream &os, const Point &p);      // Insertion
    friend std::istream &operator>>(std::istream &is, Point &p);            // Extraction
public:
    Point() = default;
    Point(double x, double y) : m_x(x), m_y(y) {}
    ~Point() = default;
    double length() const; // Function to calculate distance from the point(0,0)
private:
    double m_x{};
    double m_y{};
};

// Insertion
inline std::ostream &operator<<(std::ostream &os, const Point &p)
{
    os << "Point [ x : " << p.m_x << ", y : " << p.m_y << "]";
    return os;
}
// Extraction
inline std::istream &operator>>(std::istream &is, Point &p)
{
    double x;
    double y;
    //fmt::print("Please type in the coordinates for the point\n");
    //fmt::print("order [x,y], separated by spaces : ");
    std::cin >> x >> y;
    p.m_x = x;
    p.m_y = y;
    return is;
}
```