

```
#include <iostream>

class Point {
public:
    double x, y;
};

class Vector {
public:
    Point start, end;

    void offset(double offsetX, double offsetY){
        start.x += offsetX;
        start.y += offsetY;
        end.x += offsetX;
        end.y += offsetY;
    }

    void print(){
        std::cout << "(" << start.x << ", " << start.y << ")" -> (" << end.x << ", " << end.y << ")" << std::endl;
    }
};

int main() {
    Vector v1, v2;
    v1.start.x = 2; v1.end.x = 3;
    v1.start.y = 3; v1.end.y = 7;
    v2.start.x = 1; v2.end.x = 3;
    v2.start.y = 2; v2.end.y = 4;
    std::cout << "Before Offset" << std::endl;
    v1.print();
    v2.print();
    v1.offset(1,1.5);
    v2.offset(1,1.5);
    std::cout << "After Offset" << std::endl;
    v1.print();
    v2.print();
    return 0;
}
```

```
student@student-optiplex5090:~/Pranish$ g++ may281.cpp -o may281
student@student-optiplex5090:~/Pranish$ ./may281
Before Offset
(2, 3) -> (3, 7)
(1, 2) -> (3, 4)
After Offset
(3, 4.5) -> (4, 8.5)
(2, 3.5) -> (4, 5.5)
student@student-optiplex5090:~/Pranish$
```

```
#include <iostream>

class Point {
public:
    double x, y;
    Point() {
        x = 0.0; y = 0.0;
        std::cout << "default constructor" << std::endl;
    }
    Point(double nx, double ny) {
        x = nx; y = ny;
        std::cout << "2-parameter constructor" << std::endl;
    }
    void display() {
        std::cout << "(" << x << ", " << y << ")" << std::endl;
    }
};

int main() {
    Point p;
    Point q(2.3, 3.9);
    p.display();
    q.display();
    return 0;
}
```

```
student@student-optiplex5090:~/Pranish$ g++ constructors.cpp -o con
student@student-optiplex5090:~/Pranish$ ./con
default constructor
2-parameter constructor
(0, 0)
(2.3, 3.9)
student@student-optiplex5090:~/Pranish$ _
```

Constructors

- Method that is called when an instance is created

```
class Point {  
public:  
    double x, y;  
    Point() {  
        x = 0.0; y = 0.0; cout << "Point instance created" << endl;  
    }  
};  
  
int main() {  
    Point p; // Point instance created  
    // p.x is 0.0, p.y is 0.0  
}
```

Constructors

- Can have multiple constructors

```
class Point {  
public:  
    double x, y;  
    Point() {  
        x = 0.0; y = 0.0; cout << "default constructor" << endl;  
    }  
    Point(double nx, double ny) {  
        x = nx; y = ny; cout << "2-parameter constructor" << endl;  
    }  
};  
  
int main() {  
    Point p; // default constructor  
    // p.x is 0.0, p.y is 0.0)  
    Point q(2.0, 3.0); // 2-parameter constructor  
    // q.x is 2.0, q.y is 3.0)  
}
```