Welcome

Getting Started with BlueJ

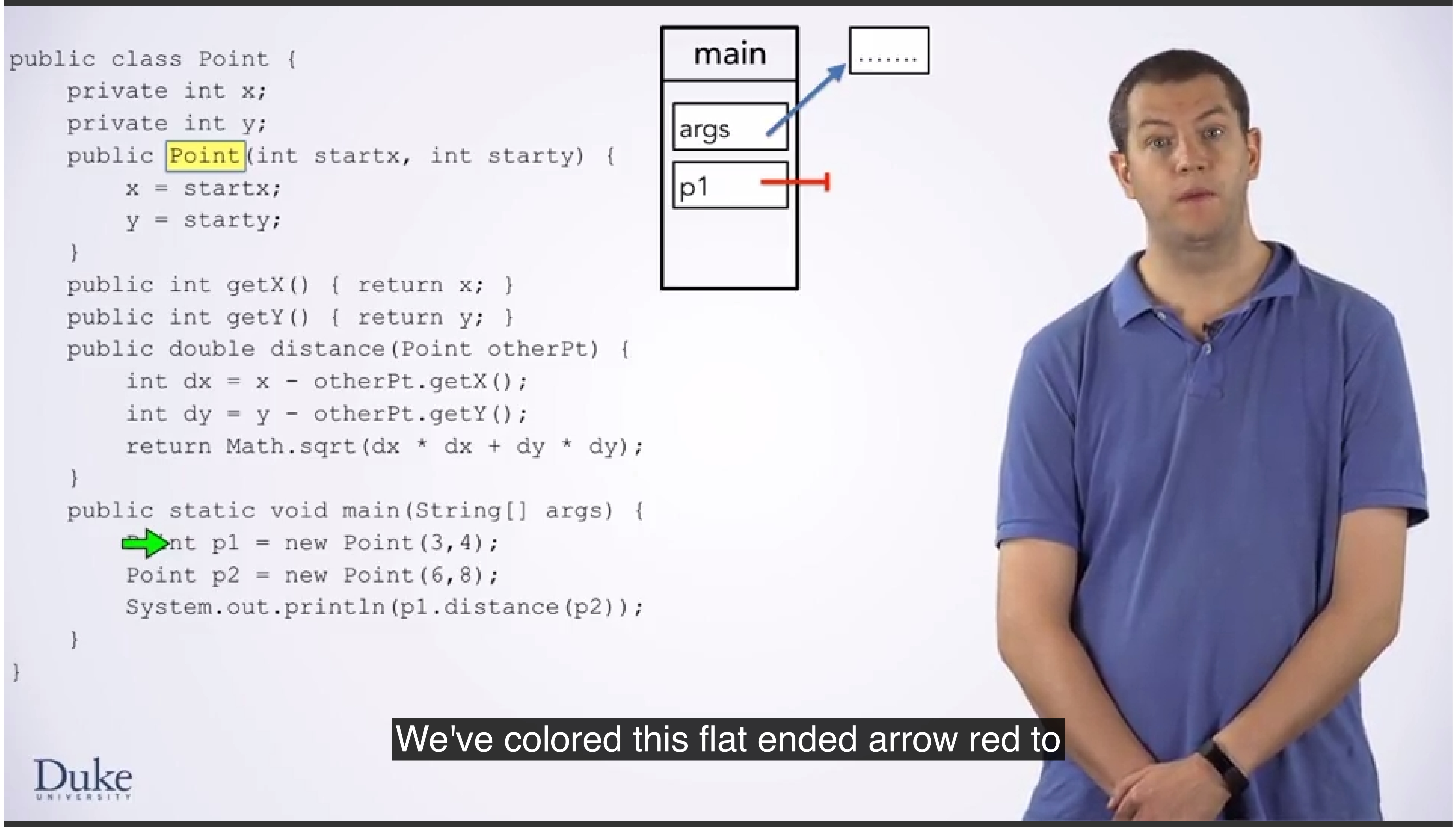
Variables and Mathematical Operators

Functions and Conditionals

Classes, Types, and For Each Loops

Classes	4 min
New	4 min
Methods	5 min
Types	5 min
For Each Loops	7 min
Practice Quiz: Classes, Types, and For Each Loops	5 questions

Seven Steps for Solving Programming Problems



```
public class Point {
    private int x;
    private int y;
    public Point(int startx, int starty) {
        x = startx;
        y = starty;
    }
    public int getX() { return x; }
    public int getY() { return y; }
    public double distance(Point otherPt) {
        int dx = x - otherPt.getX();
        int dy = y - otherPt.getY();
        return Math.sqrt(dx * dx + dy * dy);
    }
    public static void main(String[] args) {
        nt p1 = new Point(3,4);
        Point p2 = new Point(6,8);
        System.out.println(p1.distance(p2));
    }
}
```

New

Have a question? Discuss this lecture in the week forums.

Interactive Transcript

Search Transcript English

0:02
Now that you have the high level concepts, let us delve into the step by step execution of our example Point code. We're going to start in main which we'll talk about more later. Args gives you access to the command line arguments which you won't need for quite a while. So we're just going to ignore that. The first line declares point P1 and initializes it to a new point. Note that classes are types, so we can use class types that we make to declare variables. Here we want P1 to be a point. Until we finish evaluating the right hand side, we're going to draw this as an arrow with a flat end. We'll use this notation to indicate when a variable does not reference an object. Note that this is a bit different from numeric variables like ints whose initial value is zero. We've colored this flat ended arrow red to remember that we have not yet explicitly initialized this variable. Now, let us evaluate the right hand side of the initialization expression. We're doing new which will create a new object. What type of object are we going to create? We'll look right after new and see that we are creating a new point. So we're going to draw a box for a point which has fields x and y. Note that the box we just drew is not in the frame but is outside of it. It is in a different area called the heap. Anytime you use new, you create data in the heap. The important difference is that data in the heap does not go away when a function returns destroying its frame. Note that we have put zero in the field of this new point and colored them red as we have not explicitly initialized them yet. Speaking of initialization, remember that we said that the whole point of a constructor is to initialize a newly created object. The next thing that happens is that we call the constructor to initialize this point. As with any call, we set up a frame and pass parameters. However, constructors and methods take an additional implicit parameter which tells them which object they are operating on. That parameter is called this and its value is an arrow which points to the object that is being acted on. In this case, this points to the object that we are creating to tell the constructor which object to initialize. Now, we enter the code for the constructor and begin executing statements there. The first line says X gets start X. There's no X in this frame, so where do we store the value of start X? Here X refers to the field inside of this object. That is, we want to put start x's value into the X field of the object we're initializing. So to find the right box we follow the arrow for this and then look for the X field in this object and store the value three into that box. On the next line, y again refers to the field inside of this object. So we update that field to be four. Now we have finished the code inside the constructor and are ready to return to main at call site one. Back in main, we need to finish this assignment statement. To do that, we need to store the value of the right hand side into the box for P1. A new expression evaluates to an arrow pointing at the object that it created. So we'll make P1 to point at the newly made point. The next line does a similar thing. It declares P2 and initializes it to a new point. So once again we create a box for P2. We haven't initialized it explicitly yet. So we have a default value of a flattened arrow meaning it does not point to any object yet. We color red to remember that it has a default value. We then create a new point with its x and y fields set to default values of zero. And we call the constructor to initialize this point. Notice how this points at the newly created point. With multiple points in our world, it is important that we can keep track of which point we are working on. We then initialize the x inside of this point to be six as before we found the right box by following the arrow from this and we initialize the y inside of this point to be eight. Now we have finished the constructor and are again ready to return to main. In main, we finish the assignment statement setting P2 to point at the newly created object. We'll stop there and pick up with executing the method call in the next video.

Downloads

Lecture Video	mp4
Subtitles (English)	WebVTT
Transcript (English)	txt

Would you like to help us translate the transcript and subtitles into additional languages?