

Memory Allocation

```
public class Point {  
    public Point(int x, int y) {  
        px = x;  
        py = y;  
    }  
  
    public void move(int dx,  
                     int dy) {  
        px += dx;  
        py += dy;  
    }  
  
    /* instance variables */  
    private int px;  
    private int py;  
}
```

```
public class MyProgram  
    extends ConsoleProgram {  
  
    public void run() {  
        Point p1 = new Point(2, 3);  
        Point p2 = new Point(4, 5);  
        p1.move(10, 11);  
    }  
}
```

Memory Allocation

↓ heap

stack ↑

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(2, 3);
        Point p2 = new Point(4, 5);
        p1.move(10, 11);
    }
}
```

Memory Allocation

↓ heap

stack ↑

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(2, 3);
        Point p2 = new Point(4, 5);
        p1.move(10, 11);
    }
}
```



Memory Allocation

↓ heap

	<i>overhead</i>	1000
px	2	1004
py	3	1008

stack ↑

p2		FFF8
p1	1000	FFFC
	<i>overhead</i>	FFFF

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(2, 3);
        Point p2 = new Point(4, 5);
        p1.move(10, 11);
    }
}
```

Memory Allocation

↓ heap

	<i>overhead</i>	1000
px	2	1004
py	3	1008
	<i>overhead</i>	100C
px	4	1010
py	5	1014

stack ↑

p2	100C	FFF8
p1	1000	FFFC
	<i>overhead</i>	FFFF

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(2, 3);
        Point p2 = new Point(4, 5);
        p1.move(10, 11);
    }
}
```

Memory Allocation

↓ heap

	<i>overhead</i>	1000
px	2	1004
py	3	1008
	<i>overhead</i>	100C
px	4	1010
py	5	1014

stack ↑

dy	11	FFE8
dx	10	FFEC
this	1000	FFF0
	<i>overhead</i>	FFF4
p2	100C	FFF8
p1	1000	FFFC
	<i>overhead</i>	FFFF

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(2, 3);
        Point p2 = new Point(4, 5);
        p1.move(10, 11);
    }
}
```

```
public void move(int dx,
                 int dy) {

    px += dx;
    py += dy;
}
```

Memory Allocation

↓ heap

stack ↑

	overhead	1000			
px	12	1004			
py	3	1008			
	overhead	100C			
px	4	1010			
py	5	1014	dy	11	FFE8
			dx	10	FFEC
			this	1000	FFF0
				overhead	FFF4
			p2	100C	FFF8
			p1	1000	FFFC
				overhead	FFFF

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(2, 3);
        Point p2 = new Point(4, 5);
        p1.move(10, 11);
    }
}
```

```
public void move(int dx,
                 int dy) {
    px += dx;
    py += dy;
}
```

Memory Allocation

↓ heap

	<i>overhead</i>	1000
px	12	1004
py	14	1008
	<i>overhead</i>	100C
px	4	1010
py	5	1014

stack ↑

dy	11	FFE8
dx	10	FFEC
this	1000	FFF0
	<i>overhead</i>	FFF4
p2	100C	FFF8
p1	1000	FFFC
	<i>overhead</i>	FFFF

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(2, 3);
        Point p2 = new Point(4, 5);
        p1.move(10, 11);
    }
}
```

```
public void move(int dx,
                 int dy) {
    px += dx;
    py += dy;
}
```


Memory Allocation

↓ heap

stack ↑

	<i>overhead</i>	1000
px	12	1004
py	14	1008
	<i>overhead</i>	100C
px	4	1010
py	5	1014

dy	11	FFE8
dx	10	FFEC
this	1000	FFF0
	<i>overhead</i>	FFF4
p2	100C	FFF8
p1	1000	FFFC
	<i>overhead</i>	FFFF

reclaimed when
method is done

"popped off stack"

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(2, 3);
        Point p2 = new Point(4, 5);
        p1.move(10, 11);
    }
}
```

```
public void move(int dx,
                 int dy) {

    px += dx;
    py += dy;
}
```

Memory Allocation

↓ heap

	<i>overhead</i>	1000
px	12	1004
py	14	1008
	<i>overhead</i>	100C
px	4	1010
py	5	1014

stack ↑

p2	100C	FFF8
p1	1000	FFFC
	<i>overhead</i>	FFFF

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(2, 3);
        Point p2 = new Point(4, 5);
        p1.move(10, 11);
        ...
    }
}
```

Pointer Viewpoint

↓ heap

	<i>overhead</i>	1000
px	12	1004
py	14	1008
	<i>overhead</i>	100C
px	4	1010
py	5	1014

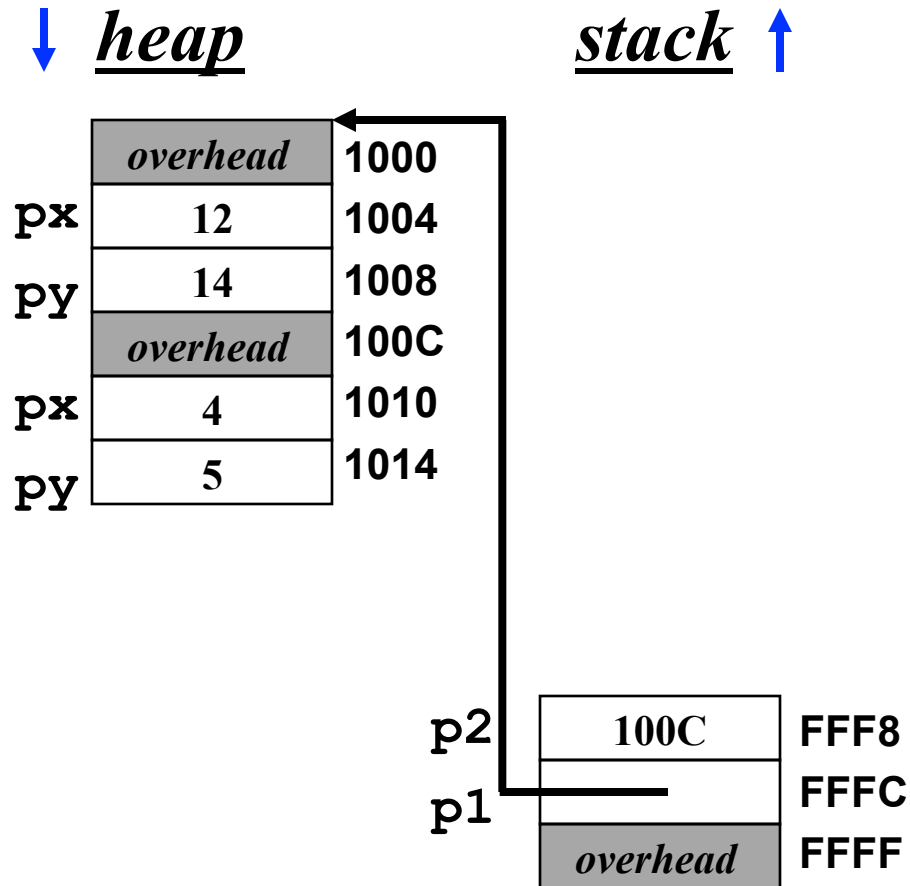
stack ↑

p2	100C	FFF8
p1	1000	FFFC
	<i>overhead</i>	FFFF

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(2, 3);
        Point p2 = new Point(4, 5);
        p1.move(10, 11);
        ...
    }
}
```

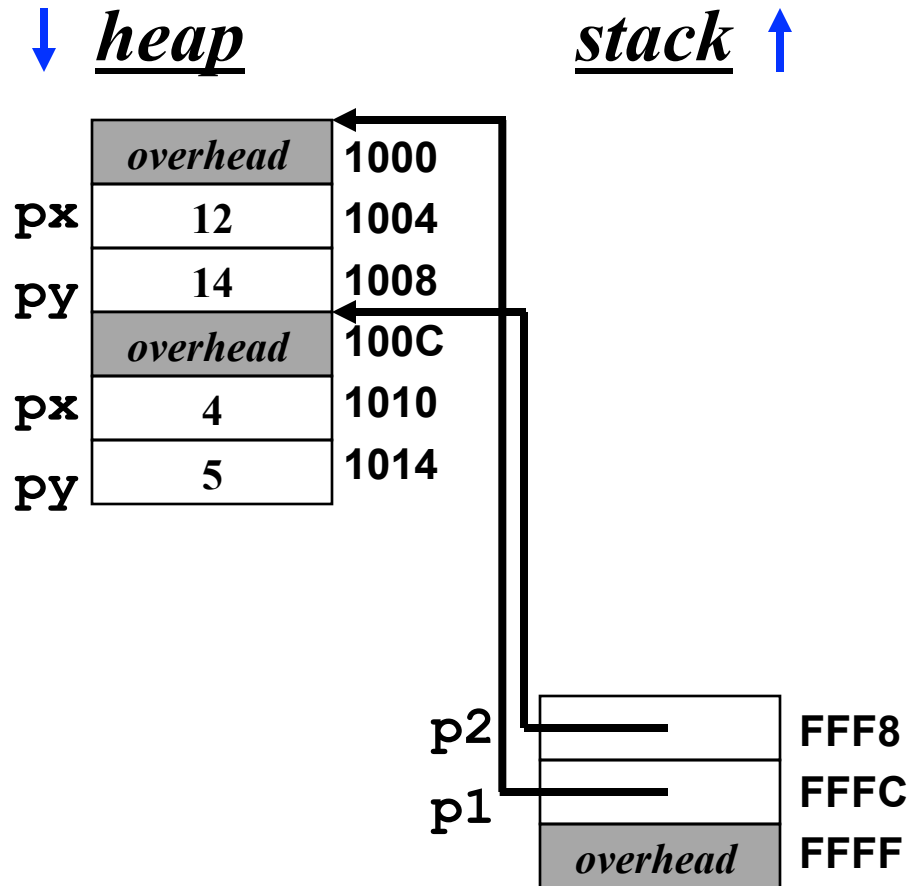
Pointer Viewpoint



```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(2, 3);
        Point p2 = new Point(4, 5);
        p1.move(10, 11);
        ...
    }
}
```

Pointer Viewpoint



```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(2, 3);
        Point p2 = new Point(4, 5);
        p1.move(10, 11);
        ...
    }
}
```

Memory Allocation

```
public class Point {  
    public Point(int x, int y) {  
        px = x;  
        py = y;  
    }  
  
    /* instance variables */  
    private int px;  
    private int py;  
}
```

```
public class Line {  
    public Line(Point p1,  
                Point p2) {  
        beg = p1;  
        end = p2;  
    }  
  
    /* instance variables */  
    private Point beg;  
    private Point end;  
}
```

```
public class MyProgram  
    extends ConsoleProgram {  
  
    public void run() {  
        Point p1 = new Point(2, 3);  
        Point p2 = new Point(4, 5);  
        Line ln = new Line(p1, p2);  
    }  
}
```

Memory Allocation

↓ heap

	<i>overhead</i>	1000
px	2	1004
py	3	1008
	<i>overhead</i>	100C
px	4	1010
py	5	1014

stack ↑

ln		FFF4
p2	100C	FFF8
p1	1000	FFFC
	<i>overhead</i>	FFFF

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(2, 3);
        Point p2 = new Point(4, 5);
        Line ln = new Line(p1, p2);
    }
}
```

Memory Allocation

↓ heap

stack ↑

	<i>overhead</i>	1000		
px	2	1004		
py	3	1008		
	<i>overhead</i>	100C		
px	4	1010		
py	5	1014		
	<i>overhead</i>	1018		
beg		101C		
end		1020	ln	FFF4
			p2	100C FFF8
			p1	1000 FFFC
				<i>overhead</i> FFFF

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(2, 3);
        Point p2 = new Point(4, 5);
        Line ln = new Line(p1, p2);
    }
}
```


Memory Allocation

↓ heap

stack ↑

	<i>overhead</i>	1000			
px	2	1004			
py	3	1008			
	<i>overhead</i>	100C			
px	4	1010	p2	100C	FFE4
py	5	1014	p1	1000	FFE8
	<i>overhead</i>	1018	this	1018	FFEC
beg		101C		<i>overhead</i>	FFF0
end		1020	ln		FFF4
			p2	100C	FFF8
			p1	1000	FFFC
				<i>overhead</i>	FFFF

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(2, 3);
        Point p2 = new Point(4, 5);
        Line ln = new Line(p1, p2);
    }
}
```

```
public Line(Point p1,
            Point p2) {
    beg = p1;
    end = p2;
}
```

Memory Allocation

↓ heap

stack ↑

	<i>overhead</i>	1000			
px	2	1004			
py	3	1008			
	<i>overhead</i>	100C			
px	4	1010	p2	100C	FFE4
py	5	1014	p1	1000	FFE8
	<i>overhead</i>	1018	this	1018	FFEC
beg	1000	101C		<i>overhead</i>	FFF0
end		1020	ln		FFF4
			p2	100C	FFF8
			p1	1000	FFFC
				<i>overhead</i>	FFFF

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(2, 3);
        Point p2 = new Point(4, 5);
        Line ln = new Line(p1, p2);
    }
}
```

```
public Line(Point p1,
            Point p2) {
    beg = p1;
    end = p2;
}
```

Memory Allocation

↓ heap

stack ↑

	overhead	1000			
px	2	1004			
py	3	1008			
	overhead	100C			
px	4	1010	p2	100C	FFE4
py	5	1014	p1	1000	FFE8
	overhead	1018	this	1018	FFEC
beg	1000	101C		overhead	FFF0
end	100C	1020	ln		FFF4
			p2	100C	FFF8
			p1	1000	FFFC
				overhead	FFFF

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(2, 3);
        Point p2 = new Point(4, 5);
        Line ln = new Line(p1, p2);
    }
}
```

```
public Line(Point p1,
            Point p2) {
    beg = p1;
    end = p2;
}
```

Memory Allocation

↓ heap

stack ↑

	<i>overhead</i>	1000		
px	2	1004		
py	3	1008		
	<i>overhead</i>	100C		
px	4	1010	p2	100C
py	5	1014	p1	1000
	<i>overhead</i>	1018	this	1018
	1000	101C		<i>overhead</i>
beg	100C	1020		FFF0
end			ln	FFF4
			p2	FFF8
			p1	FFFC
				<i>overhead</i>
				FFFF

"popped off stack"

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(2, 3);
        Point p2 = new Point(4, 5);
        Line ln = new Line(p1, p2);
    }
}
```

```
public Line(Point p1,
            Point p2) {
    beg = p1;
    end = p2;
}
```

Memory Allocation

↓ heap

	<i>overhead</i>	1000
px	2	1004
py	3	1008
	<i>overhead</i>	100C
px	4	1010
py	5	1014
	<i>overhead</i>	1018
beg	1000	101C
end	100C	1020

stack ↑

ln		FFF4
p2	100C	FFF8
p1	1000	FFFC
	<i>overhead</i>	FFFF

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(2, 3);
        Point p2 = new Point(4, 5);
        Line ln = new Line(p1, p2);
    }
}
```

Memory Allocation

↓ heap

	<i>overhead</i>	1000
px	2	1004
py	3	1008
	<i>overhead</i>	100C
px	4	1010
py	5	1014
	<i>overhead</i>	1018
beg	1000	101C
end	100C	1020

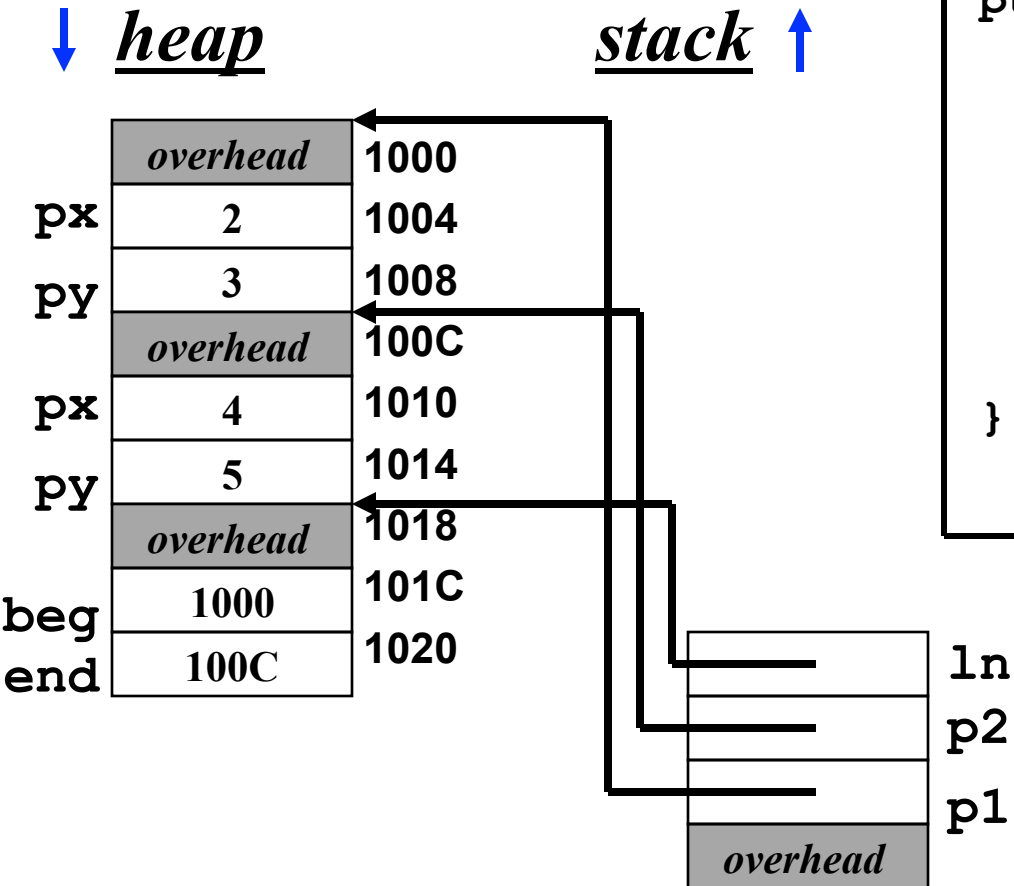
stack ↑

ln	1018	FFF4
p2	100C	FFF8
p1	1000	FFFC
	<i>overhead</i>	FFFF

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(2, 3);
        Point p2 = new Point(4, 5);
        Line ln = new Line(p1, p2);
    }
}
```

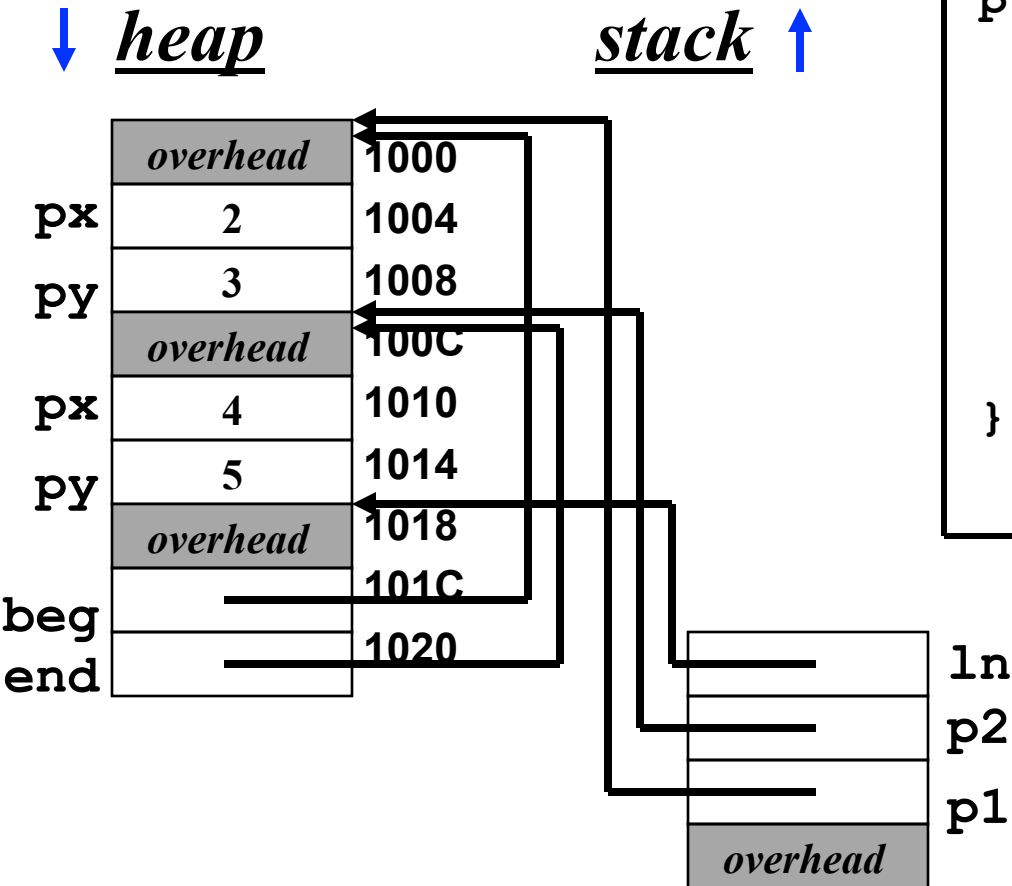
Pointer Viewpoint



```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(2, 3);
        Point p2 = new Point(4, 5);
        Line ln = new Line(p1, p2);
    }
}
```

Pointer Viewpoint



```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(2, 3);
        Point p2 = new Point(4, 5);
        Line ln = new Line(p1, p2);
    }
}
```


Another Example

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(1, 2);
        Point p2 = p1;
        p2.move(3, 4);

        Point p3;
        p3.move(1, 1);
    }
}
```

Another Example

↓ heap

stack ↑

```
public class MyProgram
    extends ConsoleProgram {

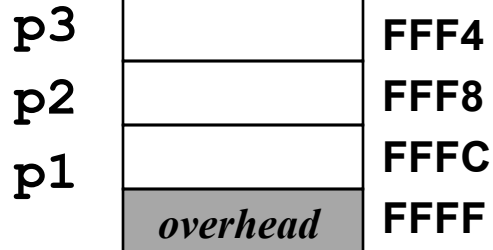
    public void run() {
        Point p1 = new Point(1, 2);
        Point p2 = p1;
        p2.move(3, 4);

        Point p3;
        p3.move(1, 1);
    }
}
```

Another Example

↓ heap

stack ↑



```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(1, 2);
        Point p2 = p1;
        p2.move(3, 4);

        Point p3;
        p3.move(1, 1);
    }
}
```

Another Example

↓ heap

	<i>overhead</i>	1000
px	1	1004
py	2	1008

stack ↑

p3		FFF4
p2		FFF8
p1	1000	FFFC
	<i>overhead</i>	FFFF

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(1, 2);
        Point p2 = p1;
        p2.move(3, 4);

        Point p3;
        p3.move(1, 1);
    }
}
```

Another Example

↓ heap

	<i>overhead</i>	1000
px	1	1004
py	2	1008

stack ↑

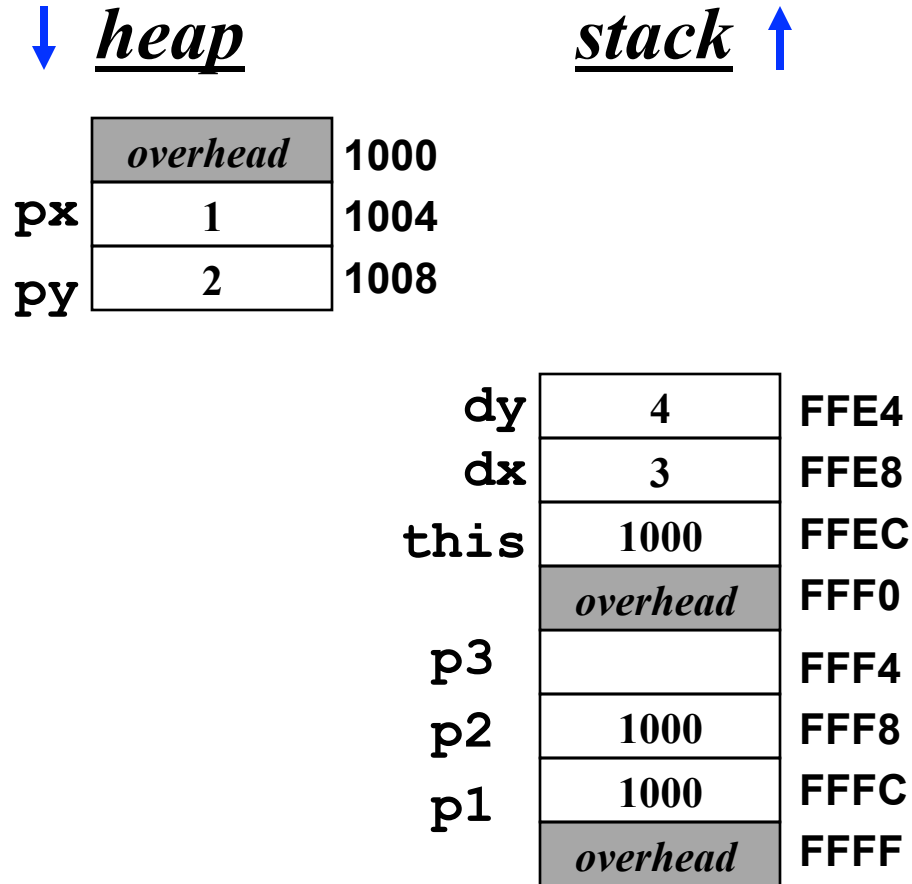
p3		FFF4
p2	1000	FFF8
p1	1000	FFFC
	<i>overhead</i>	FFFF

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(1, 2);
        Point p2 = p1;
        p2.move(3, 4);

        Point p3;
        p3.move(1, 1);
    }
}
```

Another Example



```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(1, 2);
        Point p2 = p1;
        p2.move(3, 4);

        Point p3;
        p3.move(1, 1);
    }
}
```

```
public void move(int dx,
                 int dy) {

    px += dx;
    py += dy;
}
```

Another Example

↓ heap

	<i>overhead</i>	1000
px	4	1004
py	2	1008

stack ↑

dy	4	FFE4
dx	3	FFE8
this	1000	FFEC
	<i>overhead</i>	FFF0
p3		FFF4
p2	1000	FFF8
p1	1000	FFFC
	<i>overhead</i>	FFFF

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(1, 2);
        Point p2 = p1;
        p2.move(3, 4);

        Point p3;
        p3.move(1, 1);
    }
}
```

```
public void move(int dx,
                 int dy) {
    px += dx;
    py += dy;
}
```

Another Example

↓ heap

	<i>overhead</i>	1000
px	4	1004
py	6	1008

stack ↑

dy	4	FFE4
dx	3	FFE8
this	1000	FFEC
	<i>overhead</i>	FFF0
p3		FFF4
p2	1000	FFF8
p1	1000	FFFC
	<i>overhead</i>	FFFF

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(1, 2);
        Point p2 = p1;
        p2.move(3, 4);

        Point p3;
        p3.move(1, 1);
    }
}
```

```
public void move(int dx,
                 int dy) {
    px += dx;
    py += dy;
}
```


Another Example

↓ heap

	<i>overhead</i>	1000
px	4	1004
py	6	1008

stack ↑

dy	4	FFE4
dx	3	FFE8
this	1000	FFEC
	<i>overhead</i>	FFF0
p3		FFF4
p2	1000	FFF8
p1	1000	FFFC
	<i>overhead</i>	FFFF

reclaimed when
method is done

"popped off stack"

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(1, 2);
        Point p2 = p1;
        p2.move(3, 4);

        Point p3;
        p3.move(1, 1);
    }
}
```

```
public void move(int dx,
                 int dy) {

    px += dx;
    py += dy;
}
```

Another Example

↓ heap

	<i>overhead</i>	1000
px	4	1004
py	6	1008

stack ↑

p3		FFF4
p2	1000	FFF8
p1	1000	FFFC
	<i>overhead</i>	FFFF

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(1, 2);
        Point p2 = p1;
        p2.move(3, 4);

        Point p3;
        p3.move(1, 1);
    }
}
```

Another Example

↓ heap

	<i>overhead</i>	1000
px	4	1004
py	6	1008

stack ↑

p3		FFF4
p2	1000	FFF8
p1	1000	FFFC
	<i>overhead</i>	FFFF

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(1, 2);
        Point p2 = p1;
        p2.move(3, 4);

        Point p3;
        p3.move(1, 1);
    }
}
```

Another Example

↓ heap

	<i>overhead</i>	1000
px	4	1004
py	6	1008

stack ↑

p3		FFF4
p2	1000	FFF8
p1	1000	FFFC
	<i>overhead</i>	FFFF

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(1, 2);
        Point p2 = p1;
        p2.move(3, 4);

        Point p3;
        p3.move(1, 1);
    }
}
```

Another Example

↓ heap

	<i>overhead</i>	1000
px	4	1004
py	6	1008

stack ↑

p3		FFF4
p2	1000	FFF8
p1	1000	FFFC
	<i>overhead</i>	FFFF

```
public class MyProgram
    extends ConsoleProgram {

    public void run() {
        Point p1 = new Point(1, 2);
        Point p2 = p1;
        p2.move(3, 4);

        Point p3;
        p3.move(1, 1);
    }
}
```

ERROR!

***p3 is not pointing
to a valid object!***