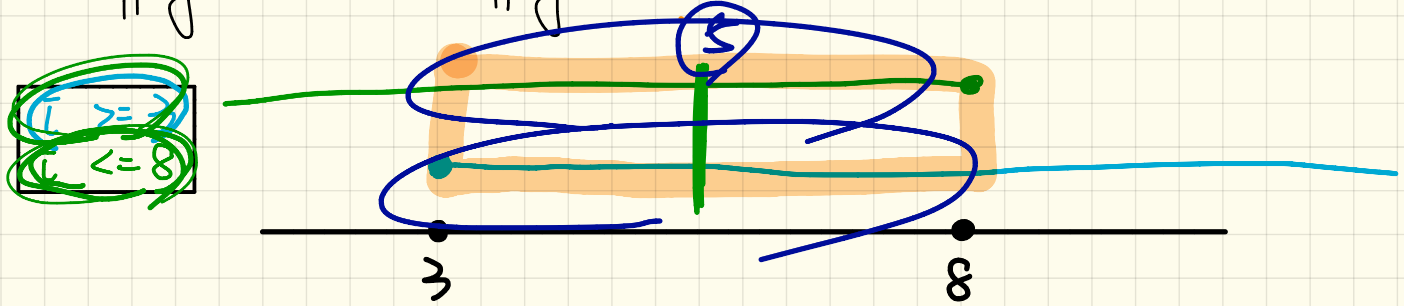
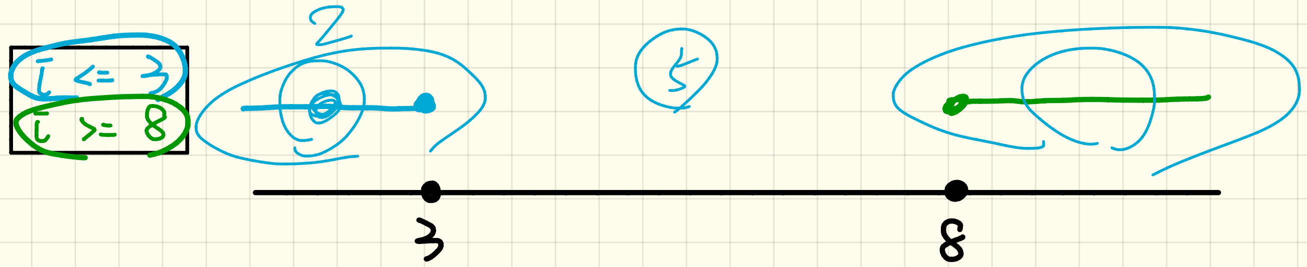
Wednesday January 23

Lecture 6

Overlapping vs. Non-Overlapping Intervals





Single If-Statement vs. Multiple If-Statements: Overlapping Conditions

a single i.s.

```
int i = 5;
if(i >= 3) {System.out.println("i is >= 3");}
else if(i <= 8) {System.out.println("i is <= 8");}
```

```
i is >= 3
```

```
int i = 5;
if(i >= 3) {System.out.println("i is >= 3");}
if(i <= 8) {System.out.println("i is <= 8");}
```

```
i is >= 3
i is <= 8
```

2 if statements.

Single If-Statement vs. Multiple If-Statements: Non-Overlapping Conditions

```java
int i = 2;
if(i <= 3) {System.out.println("i is <= 3");}
else if(i >= 8) {System.out.println("i is >= 8");}
```

1 i.S.

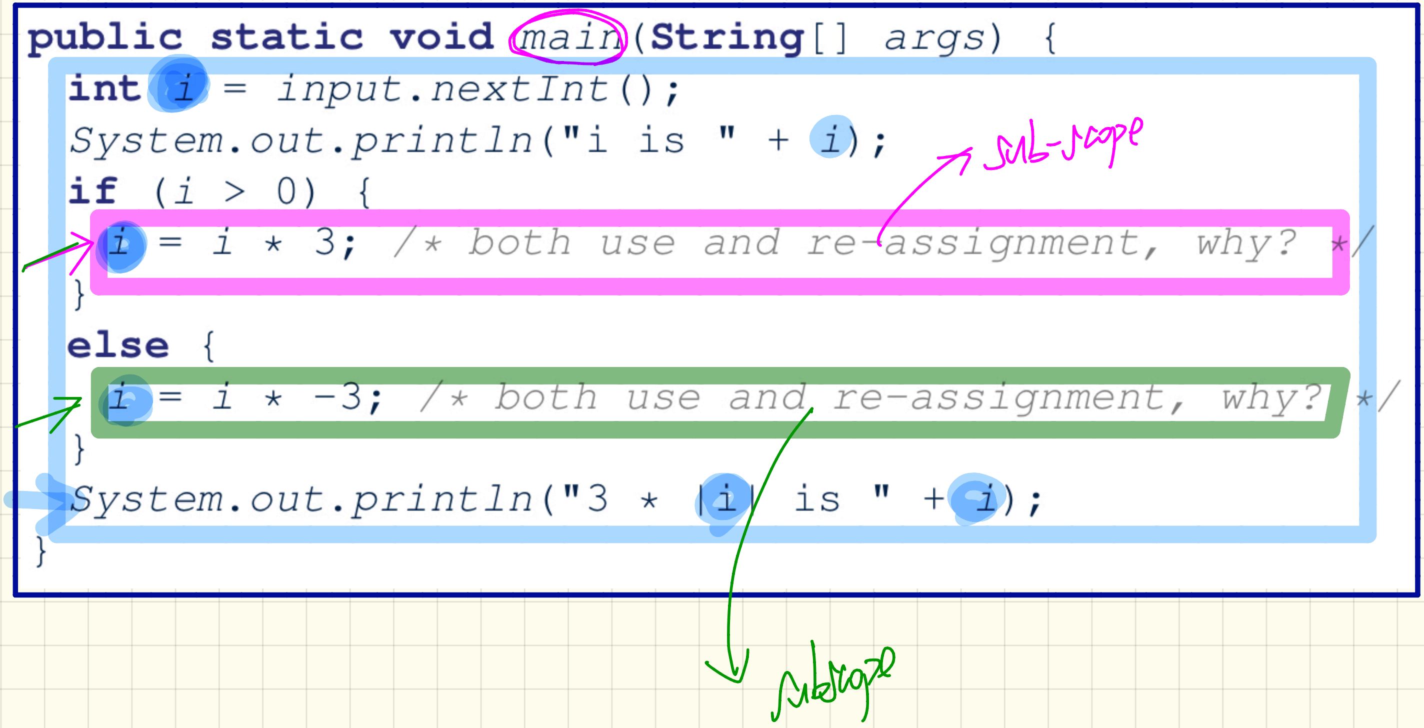
```
i is <= 3
```

```java
int i = 2;
if(i <= 3) {System.out.println("i is <= 3");}
if(i >= 8) {System.out.println("i is >= 8");}
```

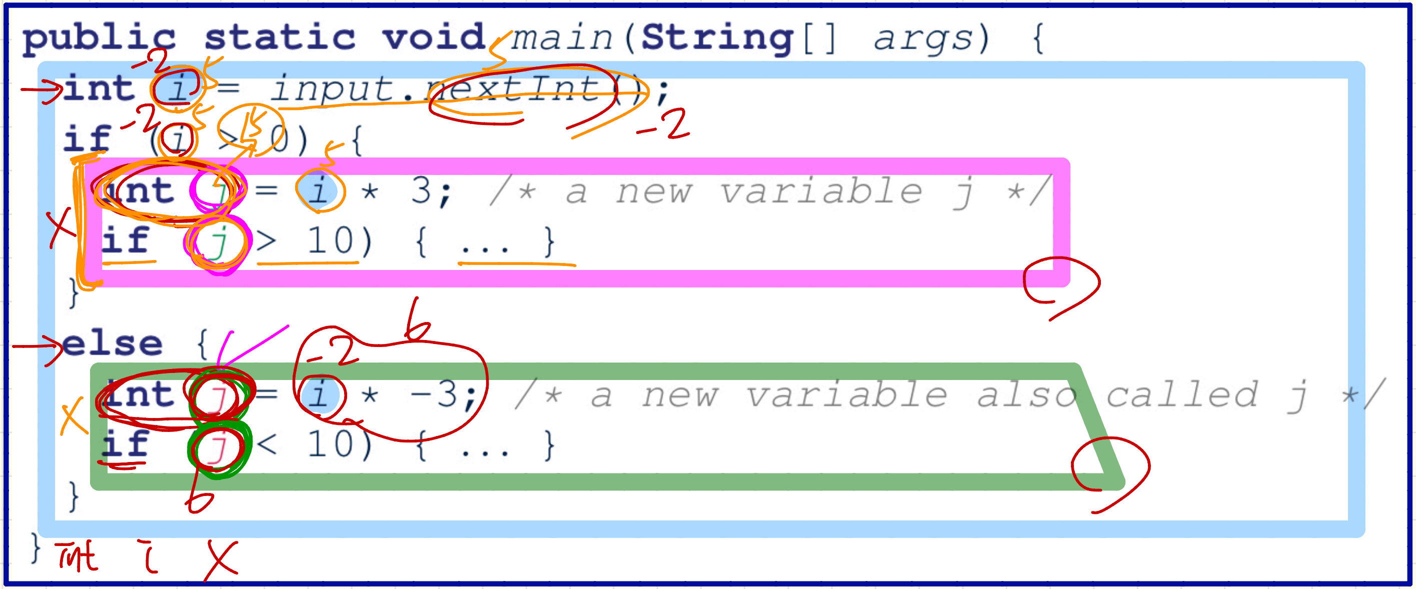
```
i is <= 3
```

2 I.S.

Scope of variables : method


```
public static void main(String[] args) {
  int i = input.nextInt();
  System.out.println("i is " + i);
  if (i > 0) {
    i = i * 3; /* both use and re-assignment, why? */
  }
  else {
    i = i * -3; /* both use and re-assignment, why? */
  }
  System.out.println("3 * |i| is " + i);
}
```

Scope of Variables : Branches



```
public static void main(String[] args) {
  int i = input.nextInt();
  if (i > 0) {
    int j = i * 3; /* a new variable j */
    if (j > 10) { ... }
  }
  else {
    int j = i * -3; /* a new variable also called j */
    if (j < 10) { ... }
  }
}
```

Scope of Variables : Illegal Use of Variable from Other Branch

```
public static void main(String[] args) {
  int i = input.nextInt();
  if (i > 0) {
    int j = i * 3; /* a new variable j */
    if (j > 10) { ... }
  }
  else {
    int k = i * -3; /* a new variable also called j */
    if (j < k) { ... }        ×
  }
}
```

```
if (k > j) { ... }
```

illegal: k is declared in a dif. sub-scope.

illegal: j is declared in a different sub-scope.

Scope of Variables: Illegal Use of Variable Outside If-Statement

```
public static void main(String[] args) {
  int i = input.nextInt();
  if (i > 0) {
    int j = i * 3; /* a new variable j */
    if (j > 10) { ... }
  }
  else {
    int j = i * -3; /* a new variable also called j */
    if (j < 10) { ... }
  }
  System.out.println("i * j is " + i * j);        ×
}
```

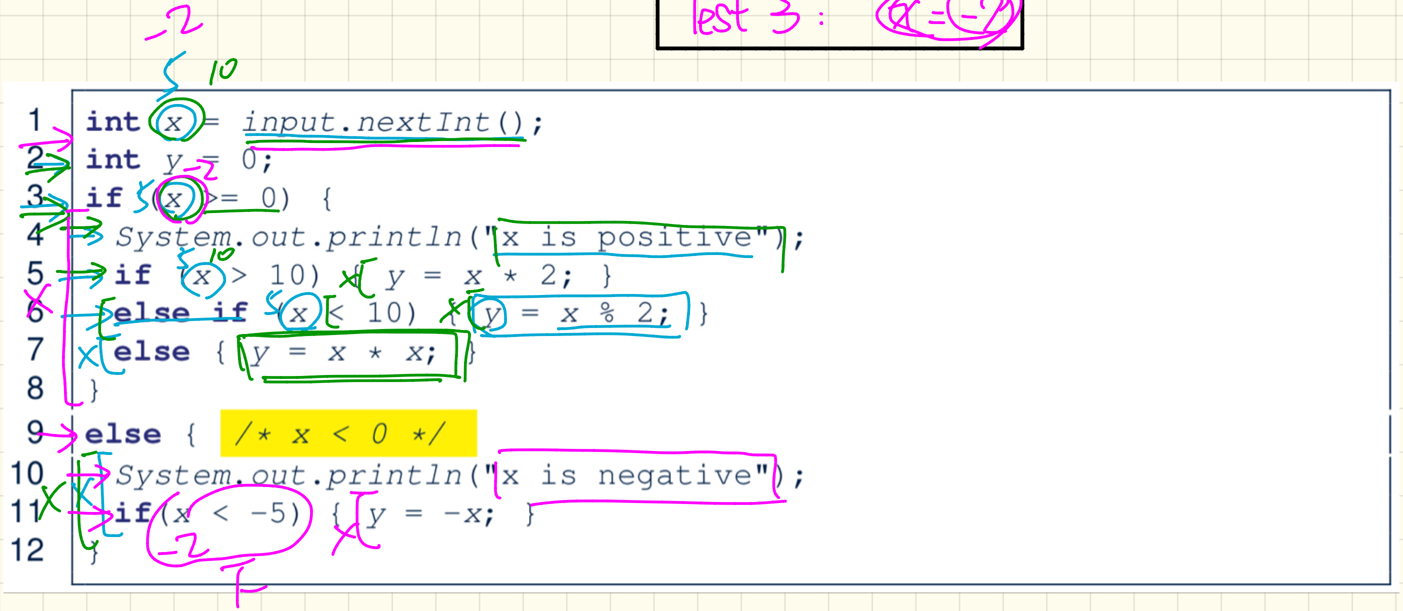
```
main(...){
 if(...){
   int j;
 }
 if(...){
   int j;
 }
}
```

Compound If-Statement

Test 1 : $x = 5$

Test 2 : $x = 10$

Test 3 : $x = -2$



```
int x = input.nextInt();
int y = 0;
if (x >= 0) {
  System.out.println("x is positive");
  if (x > 10) { y = x * 2; }
  else if (x < 10) { y = x % 2; }
  else { y = x * x; }
}
else { /* x < 0 */
  System.out.println("x is negative");
  if(x < -5) { y = -x; }
}
```

Truth Tables of Logical Operators

Negation (not)

P	!P
true	false
false	true

Conjunction (and)

P	q	P && q
false	false	
false	true	false
true	false	
true	true	true

Disjunction (or)

P	q	P || q
false	false	false
false	true	
true	false	true
true	true	

Example of Logical Operation: Negation

Test 1: 0
Test 2: -3
Test 3: 5

Operand op	!op
true	*false*
false	*true*

```
double radius = input.nextDouble();
boolean isPositive = radius > 0;
if (!isPositive) { /* not the case that isPositive is true */
  System.out.println("Error: radius value must be positive.");
}
else {
  System.out.println("Area is " + radius * radius * PI);
}
```

Example of Logical Operation: Conjunction

Test 1: age ← 30
Test 2: age ← 50
Test 3: age ← 70

Left Operand `op1`	Right Operand `op2`	`op1 && op2`
true	*true*	*true*
✓ *true*	*false*	*false*
false	*true*	*false*
✓ *false*	*false*	*false*

50 30

```
int age = input.nextInt();                      F
boolean isOldEnough = age >= 45;                 T
boolean isNotTooOld = age < 65                 T T
if (!isOldENough) { /* young */ }
else if (isOldEnough && isNotTooOld) { /* middle-aged */ }
else { /* senior */ }                T     T
```

Example of Logical Operation: Disjunction

Test 1:	age =	70
Test 2:	age =	15
Test 3:	age =	40

Left Operand `op1`	Right Operand `op2`	`op1 || op2`
false	*false*	*false*
true	*false*	*true*
false	*true*	*true*
true	*true*	*true*

70 40

```
int age = input.nextInt();
boolean isSenior = age >= 65;
boolean isChild = age < 18
if (isSenior || isChild) { /* discount */ }
else { /* no discount */ }
```

T F F F T F

F || F