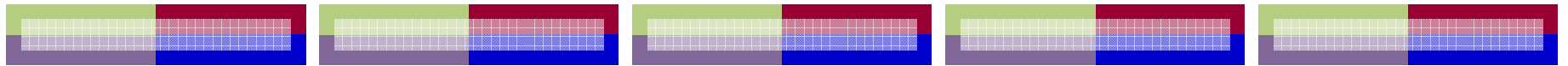


CS101

# Problem Solving and Object-Oriented Programming

## L19: For Loops and Nesting



# Counting

Initialize counter



```
int fireworkCount = 0;
```

Test counter against limit



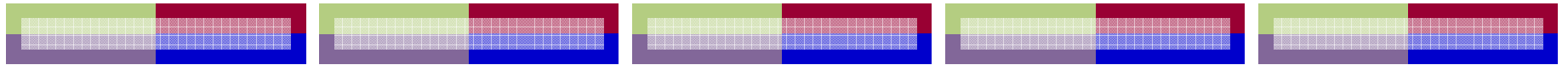
```
while (fireworkCount < NUM_IN_FINALE) {
```

```
    new Firework (...);
```

```
    fireworkCount++; ← update counter
```

```
}
```





# Counting

```
for (fireworkCount = 0;  
    fireworkCount < NUM_IN_FINALE  
    fireworkCount++){  
    new Firework(...);  
}
```

Initialize counter

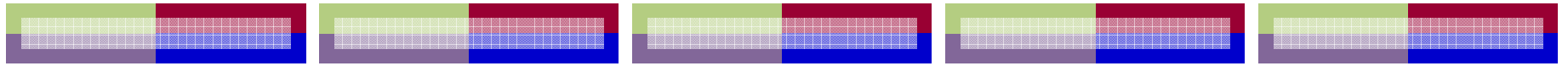
Test counter against limit

update counter

Counter update is always executed AFTER loop body

Note: counting variable can be declared before loop





# while vs for loops

initialize counter

```
while (test condition){
```

```
    loop body
```

```
    update counter
```

```
}
```

```
for ( initialize counter;
```

```
    test condition;
```

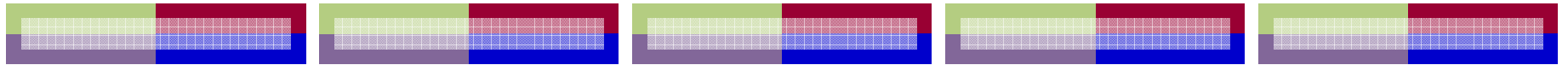
```
    update ) {
```

```
    loop body
```

```
}
```

- Initialization done once - when the loop is first reached.
- Test done before each iteration, including the first
- Update done as if it were the last statement in the loop body
- For loops are good when loop controls (initialization, test and update) all rely on the same variable, as in counting.





# Nested Loops

```
double x;  
double y;  
public void onMouseClick (Location point) {  
    // while there are more rows to knit  
    for ( y = point.getY();  
          y < point.getY() + SCARF_HEIGHT;  
          y = y + Y_DISP) {  
        // knits one row  
        for ( x = point.getX();  
              x < point.getX() + SCARF_WIDTH;  
              x = x + X_DISP) {  
            // knits one stitch  
            new FramedOval(x, y, DIAMETER, DIAMETER, canvas);  
        } // end of inner for  
    } // end of outer for  
}
```

