Data Structures and Algorithms in Java

Procedural Programming: Control Flow

Outline

If Statement

If Statement

```
if (<boolean-expression>) {
    <statement>
    ...
} else if (<boolean-expression>) {
    <statement>
    ...
} else if (<boolean-expression>) {
    <statement>
    ...
...
} else {
    <statement>
    ...
}
```

If Statement

If Statement

`Grade.java`

- Command-line input: a percentage *score* (double)
- Standard output: the corresponding letter grade

If Statement

`Grade.java`

- Command-line input: a percentage *score* (double)
- Standard output: the corresponding letter grade

```
× ~/workspace/dsaj
$ _
```

If Statement

`Grade.java`

- Command-line input: a percentage *score* (double)
- Standard output: the corresponding letter grade

```
× ~/workspace/dsaj
$ java Grade 97
```

If Statement

`Grade.java`

- Command-line input: a percentage *score* (double)
- Standard output: the corresponding letter grade

```
× ~/workspace/dsaj
$ java Grade 97
A
$ _
```

If Statement

`Grade.java`

- Command-line input: a percentage *score* (double)
- Standard output: the corresponding letter grade

```
× ~/workspace/dsaj
$ java Grade 97
A
$ java Grade 56


```

If Statement

`Grade.java`

- Command-line input: a percentage *score* (double)
- Standard output: the corresponding letter grade

× ~/workspace/dsaj

```
$ java Grade 97
A
$ java Grade 56
F
$ _
```

If Statement

If Statement

× Grade.java 1/2

```java
import stdlib.StdOut;

public class Grade {
    public static void main(String[] args) {
        double score = Double.parseDouble(args[0]);
        if (score >= 93) {
            StdOut.println("A");
        } else if (score >= 90) {
            StdOut.println("A-");
        } else if (score >= 87) {
            StdOut.println("B+");
        } else if (score >= 83) {
            StdOut.println("B");
        } else if (score >= 80) {
            StdOut.println("B-");
        } else if (score >= 77) {
            StdOut.println("C+");
        } else if (score >= 73) {
            StdOut.println("C");
        } else if (score >= 70) {
```

If Statement

If Statement

× Grade.java 2/2

```
21              StdOut.println("C-");
22          } else if (score >= 67) {
23              StdOut.println("D+");
24          } else if (score >= 63) {
25              StdOut.println("D");
26          } else if (score >= 60) {
27              StdOut.println("D-");
28          } else {
29              StdOut.println("F");
30          }
31      }
32  }
```

If Statement

If Statement

```
1  import stdlib.StdOut;
2
3  public class Grade {
4      public static void main(String[] args) {
5          double score = Double.parseDouble(args[0]);
6          if (score >= 93) {
7              StdOut.println("A");
8          } else if (score >= 90) {
9              StdOut.println("A-");
10         } else if (score >= 87) {
11             StdOut.println("B+");
12         } else if (score >= 83) {
13             StdOut.println("B");
14         } else if (score >= 80) {
15         ...
16         } else {
17             StdOut.println("F");
18         }
19     }
20 }
```

line #	score

```
1  $ _
2
3
```

If Statement

```
1  import stdlib.StdOut;
2
3  public class Grade {
4      public static void main(String[] args) {
5          double score = Double.parseDouble(args[0]);
6          if (score >= 93) {
7              StdOut.println("A");
8          } else if (score >= 90) {
9              StdOut.println("A-");
10         } else if (score >= 87) {
11             StdOut.println("B+");
12         } else if (score >= 83) {
13             StdOut.println("B");
14         } else if (score >= 80) {
15         ...
16         } else {
17             StdOut.println("F");
18         }
19     }
20 }
```

line #	score

```
1  $ java Grade 88
2
3
```

If Statement

```
1  import stdlib.StdOut;
2
3  public class Grade {
4      public static void main(String[] args) {
5          double score = Double.parseDouble(args[0]);
6          if (score >= 93) {
7              StdOut.println("A");
8          } else if (score >= 90) {
9              StdOut.println("A-");
10         } else if (score >= 87) {
11             StdOut.println("B+");
12         } else if (score >= 83) {
13             StdOut.println("B");
14         } else if (score >= 80) {
15         ...
16         } else {
17             StdOut.println("F");
18         }
19     }
20 }
```

line #	score
1	

```
1  $ java Grade 88
2
3
```

If Statement

```
1  import stdlib.StdOut;
2
3  public class Grade {
4      public static void main(String[] args) {
5          double score = Double.parseDouble(args[0]);
6          if (score >= 93) {
7              StdOut.println("A");
8          } else if (score >= 90) {
9              StdOut.println("A-");
10         } else if (score >= 87) {
11             StdOut.println("B+");
12         } else if (score >= 83) {
13             StdOut.println("B");
14         } else if (score >= 80) {
15         ...
16         } else {
17             StdOut.println("F");
18         }
19     }
20 }
```

line #	score
3	

```
1  $ java Grade 88
2
3
```

If Statement

```
1  import stdlib.StdOut;
2
3  public class Grade {
4      public static void main(String[] args) {
5          double score = Double.parseDouble(args[0]);
6          if (score >= 93) {
7              StdOut.println("A");
8          } else if (score >= 90) {
9              StdOut.println("A-");
10         } else if (score >= 87) {
11             StdOut.println("B+");
12         } else if (score >= 83) {
13             StdOut.println("B");
14         } else if (score >= 80) {
15         ...
16         } else {
17             StdOut.println("F");
18         }
19     }
20 }
```

line #	score
4	

```
1  $ java Grade 88
2
3
```

If Statement

```
1  import stdlib.StdOut;
2
3  public class Grade {
4      public static void main(String[] args) {
5          double score = Double.parseDouble(args[0]);
6          if (score >= 93) {
7              StdOut.println("A");
8          } else if (score >= 90) {
9              StdOut.println("A-");
10         } else if (score >= 87) {
11             StdOut.println("B+");
12         } else if (score >= 83) {
13             StdOut.println("B");
14         } else if (score >= 80) {
15         ...
16         } else {
17             StdOut.println("F");
18         }
19     }
20 }
```

line #	score
5	88.0

```
1  $ java Grade 88
2
3
```

If Statement

```
1  import stdlib.StdOut;
2
3  public class Grade {
4      public static void main(String[] args) {
5          double score = Double.parseDouble(args[0]);
6          if (score >= 93) {
7              StdOut.println("A");
8          } else if (score >= 90) {
9              StdOut.println("A-");
10         } else if (score >= 87) {
11             StdOut.println("B+");
12         } else if (score >= 83) {
13             StdOut.println("B");
14         } else if (score >= 80) {
15         ...
16         } else {
17             StdOut.println("F");
18         }
19     }
20 }
```

line #	score
6	88.0

```
1  $ java Grade 88
2
3
```

If Statement

```
1  import stdlib.StdOut;
2
3  public class Grade {
4      public static void main(String[] args) {
5          double score = Double.parseDouble(args[0]);
6          if (score >= 93) {
7              StdOut.println("A");
8          } else if (score >= 90) {
9              StdOut.println("A-");
10         } else if (score >= 87) {
11             StdOut.println("B+");
12         } else if (score >= 83) {
13             StdOut.println("B");
14         } else if (score >= 80) {
15         ...
16         } else {
17             StdOut.println("F");
18         }
19     }
20 }
```

line #	score
8	88.0

```
1  $ java Grade 88
2
3
```

If Statement

```
1  import stdlib.StdOut;
2
3  public class Grade {
4      public static void main(String[] args) {
5          double score = Double.parseDouble(args[0]);
6          if (score >= 93) {
7              StdOut.println("A");
8          } else if (score >= 90) {
9              StdOut.println("A-");
10         } else if (score >= 87) {
11             StdOut.println("B+");
12         } else if (score >= 83) {
13             StdOut.println("B");
14         } else if (score >= 80) {
15         ...
16         } else {
17             StdOut.println("F");
18         }
19     }
20 }
```

line #	score
10	88.0

```
1  $ java Grade 88
2
3
```

If Statement

```
1  import stdlib.StdOut;
2
3  public class Grade {
4      public static void main(String[] args) {
5          double score = Double.parseDouble(args[0]);
6          if (score >= 93) {
7              StdOut.println("A");
8          } else if (score >= 90) {
9              StdOut.println("A-");
10         } else if (score >= 87) {
11             StdOut.println("B+");
12         } else if (score >= 83) {
13             StdOut.println("B");
14         } else if (score >= 80) {
15         ...
16         } else {
17             StdOut.println("F");
18         }
19     }
20 }
```

line #	score
11	88.0

```
1  $ java Grade 88
2  B+
3
```

If Statement

```
1  import stdlib.StdOut;
2
3  public class Grade {
4      public static void main(String[] args) {
5          double score = Double.parseDouble(args[0]);
6          if (score >= 93) {
7              StdOut.println("A");
8          } else if (score >= 90) {
9              StdOut.println("A-");
10         } else if (score >= 87) {
11             StdOut.println("B+");
12         } else if (score >= 83) {
13             StdOut.println("B");
14         } else if (score >= 80) {
15         ...
16         } else {
17             StdOut.println("F");
18         }
19     }
20 }
```

line #	score

```
1  $ java Grade 88
2  B+
3  $ _
```

Conditional Expression

Conditional Expression

```
<boolean-expression> ? <then-expression> : <else-expression>
```

Conditional Expression

Conditional Expression

`Flip.java`

- Standard output: “Heads” or “Tails”

Conditional Expression

`Flip.java`

- Standard output: “Heads” or “Tails”

```
× ~/workspace/dsaj
$ _
```

Conditional Expression

`Flip.java`

- Standard output: “Heads” or “Tails”

```
× ~/workspace/dsaj
$ java Flip
```

Conditional Expression

`Flip.java`

- Standard output: “Heads” or “Tails”

```
× ~/workspace/dsaj
$ java Flip
Heads
$ _
```

Conditional Expression

`Flip.java`

- Standard output: “Heads” or “Tails”

× ~/workspace/dsaj

```
$ java Flip
Heads
$ java Flip
```

Conditional Expression

`Flip.java`

- Standard output: “Heads” or “Tails”

× ~/workspace/dsaj

```
$ java Flip
Heads
$ java Flip
Heads
$ _
```

Conditional Expression

`Flip.java`

- Standard output: “Heads” or “Tails”

× ~/workspace/dsaj

```
$ java Flip
Heads
$ java Flip
Heads
$ java Flip


```

Conditional Expression

`Flip.java`

- Standard output: “Heads” or “Tails”

× ~/workspace/dsaj

```
$ java Flip
Heads
$ java Flip
Heads
$ java Flip
Tails
$ _
```

Conditional Expression

Conditional Expression

× Flip.java

```
1  import stdlib.StdOut;
2  import stdlib.StdRandom;
3
4  public class Flip {
5      public static void main(String[] args) {
6          String result = StdRandom.bernoulli() ? "Heads" : "Tails";
7          StdOut.println(result);
8      }
9  }
```

While Statement

While Statement

```
while (<boolean-expression>) {
    <statement>
    ...
}
```

While Statement

While Statement

`NHellos.java`

- Command-line input: *n* (int)
- Standard output: *n* hellos

While Statement

NHellos.java

- Command-line input: *n* (int)
- Standard output: *n* hellos

```
× ~/workspace/dsaj
$ _
```

While Statement

NHellos.java

- Command-line input: *n* (int)
- Standard output: *n* hellos

```
× ~/workspace/dsaj
$ java NHellos 10
```

While Statement

NHellos.java

- Command-line input: *n* (int)
- Standard output: *n* hellos

```
× ~/workspace/dsaj
$ java NHellos 10
Hello # 1
Hello # 2
Hello # 3
Hello # 4
Hello # 5
Hello # 6
Hello # 7
Hello # 8
Hello # 9
Hello # 10
$ _
```

While Statement

While Statement

× NHellos.java

```java
import stdlib.StdOut;

public class NHellos {
    public static void main(String[] args) {
        int n = Integer.parseInt(args[0]);
        int i = 1;
        while (i <= n) {
            StdOut.println("Hello # " + i);
            i++;
        }
    }
}
```

While Statement

While Statement

```
1  import stdlib.StdOut;
2
3  public class NHellos {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          int i = 1;
7          while (i <= n) {
8              StdOut.println("Hello # " + i);
9              i++;
10         }
11     }
12 }
```

line #	n	i

```
$ _
```

While Statement

```
1  import stdlib.StdOut;
2
3  public class NHellos {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          int i = 1;
7          while (i <= n) {
8              StdOut.println("Hello # " + i);
9              i++;
10         }
11     }
12 }
```

line #	n	i

```
$ java NHellos 3
```

While Statement

```
1  import stdlib.StdOut;
2
3  public class NHellos {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          int i = 1;
7          while (i <= n) {
8              StdOut.println("Hello # " + i);
9              i++;
10         }
11     }
12 }
```

line #	n	i
1		

```
1  $ java NHellos 3
2
3
4
5
```

While Statement

```
1  import stdlib.StdOut;
2
3  public class NHellos {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          int i = 1;
7          while (i <= n) {
8              StdOut.println("Hello # " + i);
9              i++;
10         }
11     }
12 }
```

line #	n	i
3		

```
1  $ java NHellos 3
2
3
4
5
```

While Statement

```
1  import stdlib.StdOut;
2
3  public class NHellos {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          int i = 1;
7          while (i <= n) {
8              StdOut.println("Hello # " + i);
9              i++;
10         }
11     }
12 }
```

line #	n	i
4		

```
1  $ java NHellos 3
2
3
4
5
```

While Statement

```
1  import stdlib.StdOut;
2
3  public class NHellos {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          int i = 1;
7          while (i <= n) {
8              StdOut.println("Hello # " + i);
9              i++;
10         }
11     }
12 }
```

line #	n	i
5	3	

```
1  $ java NHellos 3
2
3
4
5
```

While Statement

```
1  import stdlib.StdOut;
2
3  public class NHellos {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          int i = 1;
7          while (i <= n) {
8              StdOut.println("Hello # " + i);
9              i++;
10         }
11     }
12 }
```

line #	n	i
6	3	1

```
1  $ java NHellos 3
2
3
4
5
```

While Statement

```
1  import stdlib.StdOut;
2
3  public class NHellos {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          int i = 1;
7          while (i <= n) {
8              StdOut.println("Hello # " + i);
9              i++;
10         }
11     }
12 }
```

line #	n	i
7	3	1

```
1  $ java NHellos 3
2
3
4
5
```

While Statement

```
1  import stdlib.StdOut;
2
3  public class NHellos {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          int i = 1;
7          while (i <= n) {
8              StdOut.println("Hello # " + i);
9              i++;
10         }
11     }
12 }
```

line #	n	i
8	3	1

```
1  $ java NHellos 3
2  Hello # 1
3
4
5
```

While Statement

```
1  import stdlib.StdOut;
2
3  public class NHellos {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          int i = 1;
7          while (i <= n) {
8              StdOut.println("Hello # " + i);
9              i++;
10         }
11     }
12 }
```

line #	n	i
9	3	2

```
1  $ java NHellos 3
2  Hello # 1
3
4
5
```

While Statement

```
1  import stdlib.StdOut;
2
3  public class NHellos {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          int i = 1;
7          while (i <= n) {
8              StdOut.println("Hello # " + i);
9              i++;
10         }
11     }
12 }
```

line #	n	i
7	3	2

```
1  $ java NHellos 3
2  Hello # 1
3
4
5
```

While Statement

```
1  import stdlib.StdOut;
2
3  public class NHellos {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          int i = 1;
7          while (i <= n) {
8              StdOut.println("Hello # " + i);
9              i++;
10         }
11     }
12 }
```

line #	n	i
8	3	2

```
1  $ java NHellos 3
2  Hello # 1
3  Hello # 2
4
5
```

While Statement

```
1  import stdlib.StdOut;
2
3  public class NHellos {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          int i = 1;
7          while (i <= n) {
8              StdOut.println("Hello # " + i);
9              i++;
10         }
11     }
12 }
```

line #	n	i
9	3	3

```
1  $ java NHellos 3
2  Hello # 1
3  Hello # 2
4
5
```

While Statement

```
1  import stdlib.StdOut;
2
3  public class NHellos {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          int i = 1;
7          while (i <= n) {
8              StdOut.println("Hello # " + i);
9              i++;
10         }
11     }
12 }
```

line #	n	i
7	3	3

```
1  $ java NHellos 3
2  Hello # 1
3  Hello # 2
4
5
```

While Statement

```
1  import stdlib.StdOut;
2
3  public class NHellos {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          int i = 1;
7          while (i <= n) {
8              StdOut.println("Hello # " + i);
9              i++;
10         }
11     }
12 }
```

line #	n	i
8	3	3

```
1  $ java NHellos 3
2  Hello # 1
3  Hello # 2
4  Hello # 3
5
```

While Statement

```
1  import stdlib.StdOut;
2
3  public class NHellos {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          int i = 1;
7          while (i <= n) {
8              StdOut.println("Hello # " + i);
9              i++;
10         }
11     }
12 }
```

line #	n	i
8	3	4

```
1  $ java NHellos 3
2  Hello # 1
3  Hello # 2
4  Hello # 3
5
```

While Statement

```
1  import stdlib.StdOut;
2
3  public class NHellos {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          int i = 1;
7          while (i <= n) {
8              StdOut.println("Hello # " + i);
9              i++;
10         }
11     }
12 }
```

line #	n	i
7	3	4

```
1  $ java NHellos 3
2  Hello # 1
3  Hello # 2
4  Hello # 3
5
```

While Statement

```
1  import stdlib.StdOut;
2
3  public class NHellos {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          int i = 1;
7          while (i <= n) {
8              StdOut.println("Hello # " + i);
9              i++;
10         }
11     }
12 }
```

line #	n	i

```
1  $ java NHellos 3
2  Hello # 1
3  Hello # 2
4  Hello # 3
5  $ _
```

For Statement

For Statement

```
for (<initialization>; <boolean-expression>; <update>) {
    <statement>
    ...
}
```

For Statement

For Statement

`Harmonic.java`

- Command-line input: n (int)
- Standard output: the nth harmonic number, $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \approx \ln(n) + 0.57721$

For Statement

`Harmonic.java`

- Command-line input: n (int)
- Standard output: the nth harmonic number, $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \approx \ln(n) + 0.57721$

```
× ~/workspace/dsaj
$ _
```

For Statement

`Harmonic.java`

- Command-line input: n (int)
- Standard output: the nth harmonic number, $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \approx \ln(n) + 0.57721$

```
× ~/workspace/dsaj
$ java Harmonic 10
```

For Statement

`Harmonic.java`

- Command-line input: n (int)
- Standard output: the nth harmonic number, $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \approx \ln(n) + 0.57721$

```
× ~/workspace/dsaj
$ java Harmonic 10
2.9289682539682538
$ _
```

For Statement

`Harmonic.java`

- Command-line input: n (int)
- Standard output: the nth harmonic number, $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \approx \ln(n) + 0.57721$

× ~/workspace/dsaj

```
$ java Harmonic 10
2.9289682539682538
$ java Harmonic 1000
```

For Statement

`Harmonic.java`

- Command-line input: n (int)
- Standard output: the nth harmonic number, $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \approx \ln(n) + 0.57721$

```
× ~/workspace/dsaj
$ java Harmonic 10
2.9289682539682538
$ java Harmonic 1000
7.485470860550343
$ _
```

For Statement

`Harmonic.java`

- Command-line input: n (int)
- Standard output: the nth harmonic number, $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \approx \ln(n) + 0.57721$

× ~/workspace/dsaj

```
$ java Harmonic 10
2.9289682539682538
$ java Harmonic 1000
7.485470860550343
$ java Harmonic 10000


```

For Statement

`Harmonic.java`

- Command-line input: n (int)
- Standard output: the nth harmonic number, $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \approx \ln(n) + 0.57721$

```
× ~/workspace/dsaj
$ java Harmonic 10
2.9289682539682538
$ java Harmonic 1000
7.485470860550343
$ java Harmonic 10000
9.787606036044348
$ _
```

For Statement

For Statement

× Harmonic.java

```java
import stdlib.StdOut;

public class Harmonic {
    public static void main(String[] args) {
        int n = Integer.parseInt(args[0]);
        double total = 0.0;
        for (int i = 1; i <= n; i++) {
            total += 1.0 / i;
        }
        StdOut.println(total);
    }
}
```

For Statement

For Statement

```
1  import stdlib.StdOut;
2
3  public class Harmonic {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          double total = 0.0;
7          for (int i = 1; i <= n; i++) {
8              total += 1.0 / i;
9          }
10         StdOut.println(total);
11     }
12 }
```

line #	n	total	i

```
1  $ _
2
3
```

For Statement

```
1  import stdlib.StdOut;
2
3  public class Harmonic {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          double total = 0.0;
7          for (int i = 1; i <= n; i++) {
8              total += 1.0 / i;
9          }
10         StdOut.println(total);
11     }
12 }
```

line #	n	total	i

```
1  $ java Harmonic 3
2
3
```

For Statement

```
1  import stdlib.StdOut;
2
3  public class Harmonic {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          double total = 0.0;
7          for (int i = 1; i <= n; i++) {
8              total += 1.0 / i;
9          }
10         StdOut.println(total);
11     }
12 }
```

line #	n	total	i
1			

```
$ java Harmonic 3
```

For Statement

```
1  import stdlib.StdOut;
2
3  public class Harmonic {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          double total = 0.0;
7          for (int i = 1; i <= n; i++) {
8              total += 1.0 / i;
9          }
10         StdOut.println(total);
11     }
12 }
```

line #	n	total	i
3			

```
1  $ java Harmonic 3
2
3
```

For Statement

```
1  import stdlib.StdOut;
2
3  public class Harmonic {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          double total = 0.0;
7          for (int i = 1; i <= n; i++) {
8              total += 1.0 / i;
9          }
10         StdOut.println(total);
11     }
12 }
```

line #	n	total	i
4			

```
1  $ java Harmonic 3
2
3
```

For Statement

```
1  import stdlib.StdOut;
2
3  public class Harmonic {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          double total = 0.0;
7          for (int i = 1; i <= n; i++) {
8              total += 1.0 / i;
9          }
10         StdOut.println(total);
11     }
12 }
```

line #	n	total	i
5	3		

```
1  $ java Harmonic 3
2
3
```

For Statement

```
1  import stdlib.StdOut;
2
3  public class Harmonic {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          double total = 0.0;
7          for (int i = 1; i <= n; i++) {
8              total += 1.0 / i;
9          }
10         StdOut.println(total);
11     }
12 }
```

line #	n	total	i
6	3	0.0	

```
1  $ java Harmonic 3
2
3
```

For Statement

```
1  import stdlib.StdOut;
2
3  public class Harmonic {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          double total = 0.0;
7          for (int i = 1; i <= n; i++) {
8              total += 1.0 / i;
9          }
10         StdOut.println(total);
11     }
12 }
```

line #	n	total	i
7	3	0.0	1

```
1  $ java Harmonic 3
2
3
```

For Statement

```
1  import stdlib.StdOut;
2
3  public class Harmonic {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          double total = 0.0;
7          for (int i = 1; i <= n; i++) {
8              total += 1.0 / i;
9          }
10         StdOut.println(total);
11     }
12 }
```

line #	n	total	i
8	3	1.0	1

```
1  $ java Harmonic 3
2
3
```

For Statement

```
1  import stdlib.StdOut;
2
3  public class Harmonic {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          double total = 0.0;
7          for (int i = 1; i <= n; i++) {
8              total += 1.0 / i;
9          }
10         StdOut.println(total);
11     }
12 }
```

line #	n	total	i
7	3	1.0	2

```
1  $ java Harmonic 3
2
3
```

For Statement

```
1  import stdlib.StdOut;
2
3  public class Harmonic {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          double total = 0.0;
7          for (int i = 1; i <= n; i++) {
8              total += 1.0 / i;
9          }
10         StdOut.println(total);
11     }
12 }
```

line #	n	total	i
8	3	1.5	2

```
1  $ java Harmonic 3
2
3
```

For Statement

```
1  import stdlib.StdOut;
2
3  public class Harmonic {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          double total = 0.0;
7          for (int i = 1; i <= n; i++) {
8              total += 1.0 / i;
9          }
10         StdOut.println(total);
11     }
12 }
```

line #	n	total	i
7	3	1.5	3

```
1  $ java Harmonic 3
2
3
```

For Statement

```
1  import stdlib.StdOut;
2
3  public class Harmonic {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          double total = 0.0;
7          for (int i = 1; i <= n; i++) {
8              total += 1.0 / i;
9          }
10         StdOut.println(total);
11     }
12 }
```

line #	n	total	i
8	3	1.83...	3

```
1  $ java Harmonic 3
2
3
```

For Statement

```
1  import stdlib.StdOut;
2
3  public class Harmonic {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          double total = 0.0;
7          for (int i = 1; i <= n; i++) {
8              total += 1.0 / i;
9          }
10         StdOut.println(total);
11     }
12 }
```

line #	n	total	i
7	3	1.83...	4

```
1  $ java Harmonic 3
2
3
```

For Statement

```
1  import stdlib.StdOut;
2
3  public class Harmonic {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          double total = 0.0;
7          for (int i = 1; i <= n; i++) {
8              total += 1.0 / i;
9          }
10         StdOut.println(total);
11     }
12 }
```

line #	n	total	i
10	3	1.83...	

```
1  $ java Harmonic 3
2  1.8333333333333333
3
```

For Statement

```
1  import stdlib.StdOut;
2
3  public class Harmonic {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          double total = 0.0;
7          for (int i = 1; i <= n; i++) {
8              total += 1.0 / i;
9          }
10         StdOut.println(total);
11     }
12 }
```

line #	n	total	i

```
1  $ java Harmonic 3
2  1.8333333333333333
3  $ _
```

Break Statement

Break Statement

```
break;
```

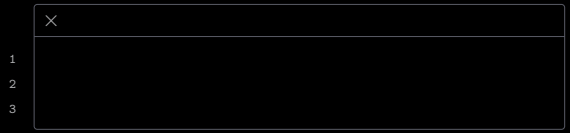
Break Statement

```
break;
```

Example

```
int i = 1;
while (true) {
    if (i >= 4) {
        break;
    }
    StdOut.println(i);
    i += 1;
}
```

line #	i



Break Statement

```
break;
```

Example

```
int i = 1;
while (true) {
    if (i >= 4) {
        break;
    }
    StdOut.println(i);
    i += 1;
}
```

line #	i
1	1

×

```
1
2
3
```

Break Statement

```
break;
```

Example

```
int i = 1;
while (true) {
    if (i >= 4) {
        break;
    }
    StdOut.println(i);
    i += 1;
}
```

line #	i
2	1

×

```
1
2
3
```

Break Statement

```
break;
```

Example

```
int i = 1;
while (true) {
    if (i >= 4) {
        break;
    }
    StdOut.println(i);
    i += 1;
}
```

line #	i
3	1

×

```
1
2
3
```

Break Statement

```
break;
```

Example

```
int i = 1;
while (true) {
    if (i >= 4) {
        break;
    }
    StdOut.println(i);
    i += 1;
}
```

line #	i
6	1

×

```
1
2
3
```

Break Statement

```
break;
```

Example

```
int i = 1;
while (true) {
    if (i >= 4) {
        break;
    }
    StdOut.println(i);
    i += 1;
}
```

line #	i
7	2

×

```
1
2
3
```

Break Statement

```
break;
```

Example

```
int i = 1;
while (true) {
    if (i >= 4) {
        break;
    }
    StdOut.println(i);
    i += 1;
}
```

line #	i
2	2

×

```
1
2
3
```

Break Statement

```
break;
```

Example

```
int i = 1;
while (true) {
    if (i >= 4) {
        break;
    }
    StdOut.println(i);
    i += 1;
}
```

line #	i
3	2

×

```
1
2
3
```

Break Statement

```
break;
```

Example

```
int i = 1;
while (true) {
    if (i >= 4) {
        break;
    }
    StdOut.println(i);
    i += 1;
}
```

line #	i
6	2

×

```
1
2
```

Break Statement

```
break;
```

Example

```
int i = 1;
while (true) {
    if (i >= 4) {
        break;
    }
    StdOut.println(i);
    i += 1;
}
```

line #	i
7	3

×

```
1
2
```

Break Statement

```
break;
```

Example

```
int i = 1;
while (true) {
    if (i >= 4) {
        break;
    }
    StdOut.println(i);
    i += 1;
}
```

line #	i
2	3

×

```
1
2
```

Break Statement

```
break;
```

Example

```
int i = 1;
while (true) {
    if (i >= 4) {
        break;
    }
    StdOut.println(i);
    i += 1;
}
```

line #	i
3	3

×

```
1
2
```

Break Statement

```
break;
```

Example

```
int i = 1;
while (true) {
    if (i >= 4) {
        break;
    }
    StdOut.println(i);
    i += 1;
}
```

line #	i
6	3

×

```
1
2
3
```

Break Statement

```
break;
```

Example

```
int i = 1;
while (true) {
    if (i >= 4) {
        break;
    }
    StdOut.println(i);
    i += 1;
}
```

line #	i
7	4

×

```
1
2
3
```

Break Statement

```
break;
```

Example

```
int i = 1;
while (true) {
    if (i >= 4) {
        break;
    }
    StdOut.println(i);
    i += 1;
}
```

line #	i
2	4

×

```
1
2
3
```

Break Statement

```
break;
```

Example

```
int i = 1;
while (true) {
    if (i >= 4) {
        break;
    }
    StdOut.println(i);
    i += 1;
}
```

line #	i
3	4

×

```
1
2
3
```

Break Statement

```
break;
```

Example

```
int i = 1;
while (true) {
    if (i >= 4) {
        break;
    }
    StdOut.println(i);
    i += 1;
}
```

line #	i
4	4

×

```
1
2
3
```

Break Statement

```
break;
```

Example

```
int i = 1;
while (true) {
    if (i >= 4) {
        break;
    }
    StdOut.println(i);
    i += 1;
}
```

line #	i

×

```
1
2
3
```

Continue Statement

Continue Statement

```
continue;
```

Continue Statement

```
continue;
```

Example

```
for (int i = 1; i <= 6; i++) {
    if (i % 2 != 0) {
        continue;
    }
    StdOut.println(i);
}
```

line #	i

×

```
1
2
3
```

Continue Statement

```
continue;
```

Example

```
for (int i = 1; i <= 6; i++) {
    if (i % 2 != 0) {
        continue;
    }
    StdOut.println(i);
}
```

line #	i
1	1

×

```
1
2
3
```

Continue Statement

```
continue;
```

Example

```
for (int i = 1; i <= 6; i++) {
    if (i % 2 != 0) {
        continue;
    }
    StdOut.println(i);
}
```

line #	i
2	1

×

```
1
2
3
```

Continue Statement

```
continue;
```

Example

```
for (int i = 1; i <= 6; i++) {
    if (i % 2 != 0) {
        continue;
    }
    StdOut.println(i);
}
```

line #	i
3	1

×

```
1
2
3
```

Continue Statement

```
continue;
```

Example

```
for (int i = 1; i <= 6; i++) {
    if (i % 2 != 0) {
        continue;
    }
    StdOut.println(i);
}
```

line #	i
1	2

×

```
1
2
3
```

Continue Statement

```
continue;
```

Example

```
for (int i = 1; i <= 6; i++) {
    if (i % 2 != 0) {
        continue;
    }
    StdOut.println(i);
}
```

line #	i
2	2

×

```
1
2
3
```

Continue Statement

```
continue;
```

Example

```
for (int i = 1; i <= 6; i++) {
    if (i % 2 != 0) {
        continue;
    }
    StdOut.println(i);
}
```

line #	i
5	2

×

```
2
```

Continue Statement

```
continue;
```

Example

```
for (int i = 1; i <= 6; i++) {
    if (i % 2 != 0) {
        continue;
    }
    StdOut.println(i);
}
```

line #	i
1	3

×

```
2
```

Continue Statement

```
continue;
```

Example

```
for (int i = 1; i <= 6; i++) {
    if (i % 2 != 0) {
        continue;
    }
    StdOut.println(i);
}
```

line #	i
2	3

×

```
2
```

Continue Statement

```
continue;
```

Example

```
for (int i = 1; i <= 6; i++) {
    if (i % 2 != 0) {
        continue;
    }
    StdOut.println(i);
}
```

line #	i
3	3

×

```
2
```

Continue Statement

```
continue;
```

Example

```
for (int i = 1; i <= 6; i++) {
    if (i % 2 != 0) {
        continue;
    }
    StdOut.println(i);
}
```

line #	i
1	4

×

```
2
```

Continue Statement

```
continue;
```

Example

```
for (int i = 1; i <= 6; i++) {
    if (i % 2 != 0) {
        continue;
    }
    StdOut.println(i);
}
```

line #	i
2	4

×

```
2
```

Continue Statement

```
continue;
```

Example

```
for (int i = 1; i <= 6; i++) {
    if (i % 2 != 0) {
        continue;
    }
    StdOut.println(i);
}
```

line #	i
5	4

×

```
2
4

```

Continue Statement

```
continue;
```

Example

```
for (int i = 1; i <= 6; i++) {
    if (i % 2 != 0) {
        continue;
    }
    StdOut.println(i);
}
```

line #	i
1	5

×

```
2
4

```

Continue Statement

```
continue;
```

Example

```
for (int i = 1; i <= 6; i++) {
    if (i % 2 != 0) {
        continue;
    }
    StdOut.println(i);
}
```

line #	i
2	5

×

```
2
4

```

Continue Statement

```
continue;
```

Example

```
for (int i = 1; i <= 6; i++) {
    if (i % 2 != 0) {
        continue;
    }
    StdOut.println(i);
}
```

line #	i
3	5

×

```
2
4

```

Continue Statement

```
continue;
```

Example

```
for (int i = 1; i <= 6; i++) {
    if (i % 2 != 0) {
        continue;
    }
    StdOut.println(i);
}
```

line #	i
1	6

×

```
2
4

```

Continue Statement

```
continue;
```

Example

```
for (int i = 1; i <= 6; i++) {
    if (i % 2 != 0) {
        continue;
    }
    StdOut.println(i);
}
```

line #	i
2	6

×

```
2
4

```

Continue Statement

```
continue;
```

Example

```
for (int i = 1; i <= 6; i++) {
    if (i % 2 != 0) {
        continue;
    }
    StdOut.println(i);
}
```

line #	i
5	6

×

```
2
4
6
```

Continue Statement

```
continue;
```

Example

```
for (int i = 1; i <= 6; i++) {
    if (i % 2 != 0) {
        continue;
    }
    StdOut.println(i);
}
```

line #	i
1	7

×

```
2
4
6
```

Continue Statement

```
continue;
```

Example

```
for (int i = 1; i <= 6; i++) {
    if (i % 2 != 0) {
        continue;
    }
    StdOut.println(i);
}
```

line #	i

×

```
2
4
6
```

Nesting

Nesting

If, while, and for statements can be nested within one another

Nesting

Nesting

`DivisorPattern.java`

- Command-line input: n (int)
- Standard output: a table where entry (i, j) is a star ("*") if j divides i or i divides j and a space (" ") otherwise

Nesting

`DivisorPattern.java`

- Command-line input: n (int)
- Standard output: a table where entry (i, j) is a star (“*”) if j divides i or i divides j and a space (“ ”) otherwise

```
× ~/workspace/dsaj
$ _
```

Nesting

`DivisorPattern.java`

- Command-line input: n (int)
- Standard output: a table where entry (i, j) is a star (“*”) if j divides i or i divides j and a space (“ ”) otherwise

```
× ~/workspace/dsaj
$ java DivisorPattern 5
1
2
3
4
5
6
7
```

Nesting

`DivisorPattern.java`

- Command-line input: n (int)
- Standard output: a table where entry (i, j) is a star ("*") if j divides i or i divides j and a space (" ") otherwise

~/workspace/dsaj

```
$ java DivisorPattern 5
* * * * * 1
* *   *   2
*   *     3
* *   *   4
*       * 5
$ _
```

Nesting

Nesting

× DivisorPattern.java

```java
import stdlib.StdOut;

public class DivisorPattern {
    public static void main(String[] args) {
        int n = Integer.parseInt(args[0]);
        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <= n; j++) {
                if (i % j == 0 || j % i == 0) {
                    StdOut.print("* ");
                } else {
                    StdOut.print("  ");
                }
            }
            StdOut.println(i);
        }
    }
}
```

Nesting

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j

```
$ _
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j

```
1  $ java DivisorPattern 3
2
3
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
1			

×

```
1  $ java DivisorPattern 3
2
3
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
3			

×

```
1  $ java DivisorPattern 3
2
3
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
4			

```
1  $ java DivisorPattern 3
2
3
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
5	3		

```
1  $ java DivisorPattern 3
2
3
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
6	3	1	

```
1  $ java DivisorPattern 3
2
3
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
7	3	1	1

```
1  $ java DivisorPattern 3
2
3
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
8	3	1	1

```
1  $ java DivisorPattern 3
2
3
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
9	3	1	1

```
1  $ java DivisorPattern 3
2  *
3
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
7	3	1	2

```
1  $ java DivisorPattern 3
2  *
3
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
8	3	1	2

```
1  $ java DivisorPattern 3
2  *
3
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
9	3	1	2

```
1  $ java DivisorPattern 3
2  * *
3
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
7	3	1	3

```
1  $ java DivisorPattern 3
2  * *
3
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
8	3	1	3

```
1  $ java DivisorPattern 3
2  * *
3
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
9	3	1	3

```
1  $ java DivisorPattern 3
2  * * *
3
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
7	3	1	4

```
1  $ java DivisorPattern 3
2  * * *
3
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
14	3	1	

```
1  $ java DivisorPattern 3
2  * * * 1
3
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
6	3	2	

```
1  $ java DivisorPattern 3
2  * * * 1
3
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
7	3	2	1

```
1  $ java DivisorPattern 3
2  * * * 1
3
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
8	3	2	1

```
1  $ java DivisorPattern 3
2  * * * 1
3
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
9	3	2	1

```
1  $ java DivisorPattern 3
2  * * * 1
3  *
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
7	3	2	2

```
1  $ java DivisorPattern 3
2  * * * 1
3  *
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
8	3	2	2

```
1  $ java DivisorPattern 3
2  * * * 1
3  *
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
9	3	2	2

```
1  $ java DivisorPattern 3
2  * * * 1
3  * *
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
7	3	2	3

```
1  $ java DivisorPattern 3
2  * * * 1
3  * *
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
8	3	2	3

```
1  $ java DivisorPattern 3
2  * * * 1
3  * *
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
11	3	2	3

```
1  $ java DivisorPattern 3
2  * * * 1
3  * *
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
7	3	2	4

```
1  $ java DivisorPattern 3
2  * * * 1
3  * *
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
14	3	2	

```
1  $ java DivisorPattern 3
2  * * * 1
3  * *   2
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
6	3	3	

```
1  $ java DivisorPattern 3
2  * * * 1
3  * *   2
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
7	3	3	1

```
1  $ java DivisorPattern 3
2  * * * 1
3  * *   2
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
8	3	3	1

```
1  $ java DivisorPattern 3
2  * * * 1
3  * *   2
4
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
9	3	3	1

```
1  $ java DivisorPattern 3
2  * * * 1
3  * *   2
4  *
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
7	3	3	2

```
1  $ java DivisorPattern 3
2  * * * 1
3  * *   2
4  *
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
8	3	3	2

```
1  $ java DivisorPattern 3
2  * * * 1
3  * *   2
4  *
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
11	3	3	2

```
1  $ java DivisorPattern 3
2  * * * 1
3  * *   2
4  *
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
7	3	3	3

```
1  $ java DivisorPattern 3
2  * * * 1
3  * *   2
4  *
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
8	3	3	3

```
1  $ java DivisorPattern 3
2  * * * 1
3  * *   2
4  *
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
9	3	3	3

```
1  $ java DivisorPattern 3
2  * * * 1
3  * *   2
4  *   *
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
7	3	3	4

```
1  $ java DivisorPattern 3
2  * * * 1
3  * *   2
4  *   *
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
14	3	3	

```
1  $ java DivisorPattern 3
2  * * * 1
3  * *   2
4  *   * 3
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j
6	3	4	

```
1  $ java DivisorPattern 3
2  * * * 1
3  * *   2
4  *   * 3
5
```

Nesting

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

line #	n	i	j

```
1  $ java DivisorPattern 3
2  * * * 1
3  * *   2
4  *   * 3
5  $ _
```

Variable Scope

Variable Scope

Part of the program that can refer to the variable by name

Variable Scope

Part of the program that can refer to the variable by name

Example

× DivisorPattern.java

```
1  import stdlib.StdOut;
2
3  public class DivisorPattern {
4      public static void main(String[] args) {
5          int n = Integer.parseInt(args[0]);
6          for (int i = 1; i <= n; i++) {
7              for (int j = 1; j <= n; j++) {
8                  if (i % j == 0 || j % i == 0) {
9                      StdOut.print("* ");
10                 } else {
11                     StdOut.print("  ");
12                 }
13             }
14             StdOut.println(i);
15         }
16     }
17 }
```

Variable	Scope
args	lines 4 — 16
n	lines 5 — 16
i	lines 6 — 15
j	lines 7 — 13