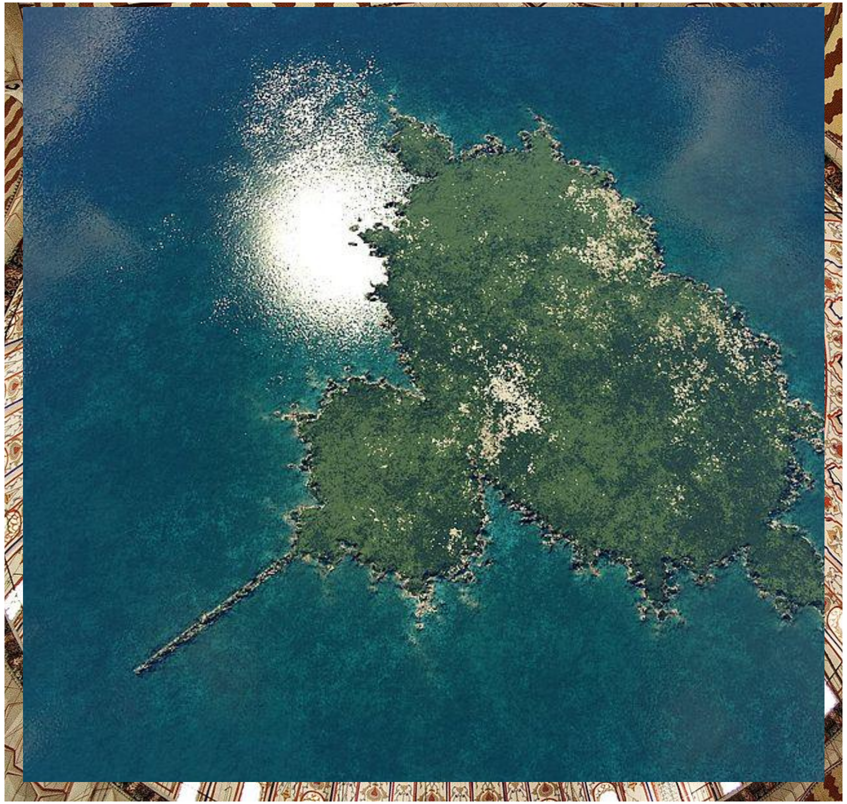
Unit 7 - Lesson 7
Recursion

Computer Science A



Fractal art is created by repeating simple patterns. It can be found in:

- computer-generated art
- architecture
- computer-generated landscapes and scenery

💡 Discuss:

How is **repetition** used in fractal art?

🚀 Question of the Day

What is recursion?

```
public int recursiveSum(int num) {
  if (num <= 1) {
     return num;
  }


  return num + recursiveSum(num - 1);
}
```

The **base case** is the instance where a recursive method will return a value rather than calling itself.

The **recursive case** is the instance where the recursive method calls itself.

Recursion is when a method calls itself.

📝 **Unit 7 Guide**

The **base case** comes from the part of the iterative method that **stops the repetition**. It occurs when a **certain condition is met**.

```
public int recursiveSum(int num) {
    int sum = 0;

    for (int i = num; i > 0; i--) {
        sum += i;
    }

    return sum;
}
```

```
public int recursiveSum(int num) {
   if (num <= 1) {
       return num;
   }

   return num + recursiveSum(num - 1);
}
```

📝 **Unit 7 Guide**

The **recursive case** from from the part of the iterative method that **repeats**.

```
public int recursiveSum(int num) {
   int sum = 0;

   for (int i = num; i > 0; i--) {
      sum += i;
   }

   return sum;
}
```

```
public int recursiveSum(int num) {
  if (num <= 1) {
     return num;
  }

  return num + recursiveSum(num - 1);
}
```

```
public int recursiveSum(int num) {
  if (num <= 1) {
     return num;
  }

  return num + recursiveSum(num - 1);
}
```

The **parameter values** capture the **progress of a recursive process**, just like how **loop control variables** capture the **progress of a loop**.

The **recursive call** has its own set of **local variables**, including the **formal parameters**.

🎥 Recursion

When is recursion useful?

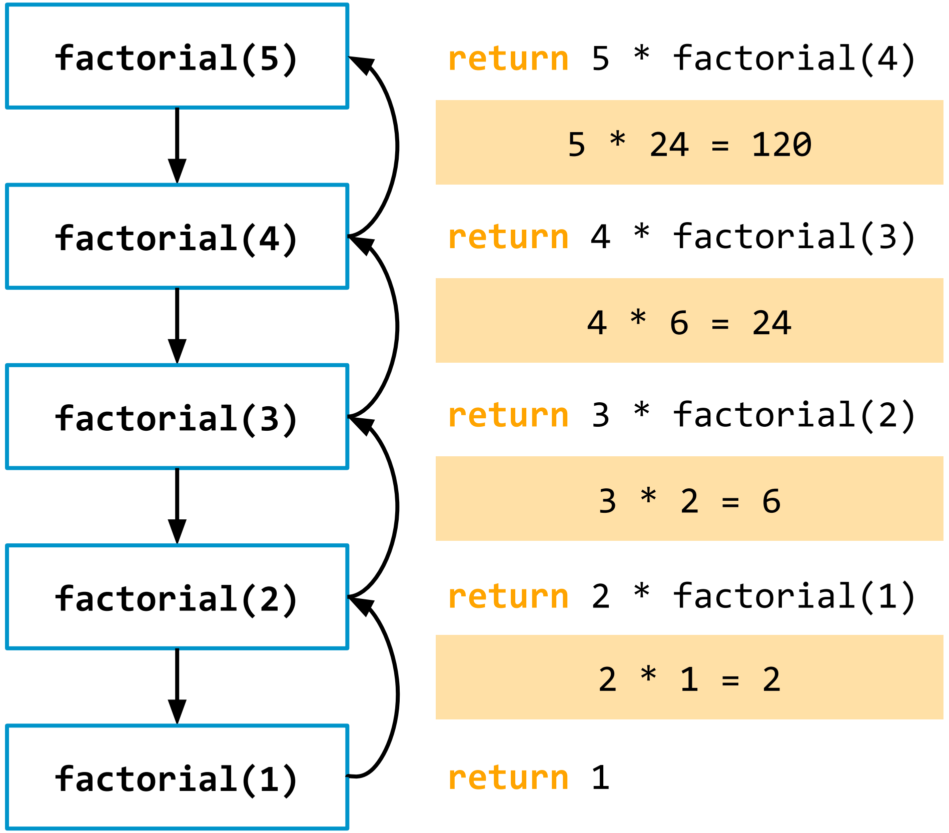
Complete the guided notes on the 📝 **Unit 7 Guide**.

Factorial.java

```
public int factorial(int n) {
   if (n == 1) {
      return 1;
   }
   else {
      return n * factorial(n - 1);
   }
}
```

What is the result of `factorial(5)`?

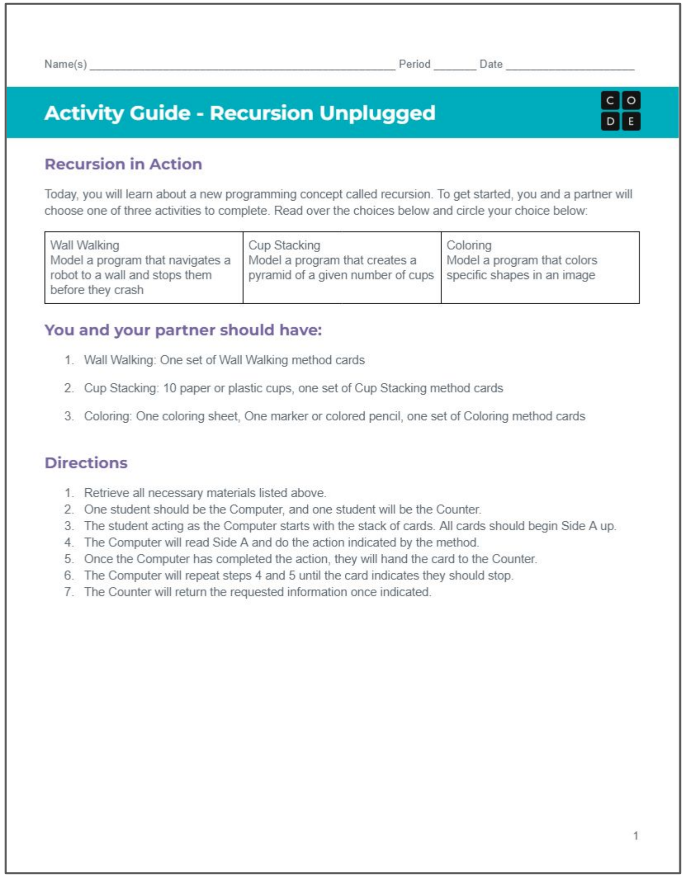
120



Unit 7 Guide

✅ Do This:

Complete **Part B** of the Recursion Unplugged handout.



Name(s) ______________________________ Period ______ Date ______________

Activity Guide - Recursion Unplugged

Recursion in Action

Today, you will learn about a new programming concept called recursion. To get started, you and a partner will choose one of three activities to complete. Read over the choices below and circle your choice below:

Wall Walking Model a program that navigates a robot to a wall and stops them before they crash	Cup Stacking Model a program that creates a pyramid of a given number of cups	Coloring Model a program that colors specific shapes in an image

You and your partner should have:

1. Wall Walking: One set of Wall Walking method cards
2. Cup Stacking: 10 paper or plastic cups, one set of Cup Stacking method cards
3. Coloring: One coloring sheet, One marker or colored pencil, one set of Coloring method cards

Directions

1. Retrieve all necessary materials listed above.
2. One student should be the Computer, and one student will be the Counter.
3. The student acting as the Computer starts with the stack of cards. All cards should begin Side A up.
4. The Computer will read Side A and do the action indicated by the method.
5. Once the Computer has completed the action, they will hand the card to the Counter.
6. The Computer will repeat steps 4 and 5 until the card indicates they should stop.
7. The Counter will return the requested information once indicated.

1

💼 Key Vocabulary

- **base case:** the instance where a recursive method will return a value rather than calling itself
- **recursion:** when a method calls itself
- **recursive case:** the instance where a recursive method calls itself