Four Core Ideas

In JAVA
What Are They?

Some parts of JAVA will be brought up again and again throughout your programming experiences. Among others, recursion, overloading methods, nested loops, and switch statements are all particularly useful in different ways.
1. Recursion

Recursion is the process of a method calling itself.

What this does is allows the coder to store information on the stack while the program does repetitive processes.

The stack is a data structure implemented by JAVA that stores data in a first on-last off manner (like a deck of cards. Keep putting cards on top of the deck and then remove them from the top in that order).
How can we apply this?

Here is a couple of methods you can try yourself:

```java
public static void recursion1(int i){
    if( i < 0)
        return;
    System.out.print(i + " ");
    recursion1(i - 1);
}

public static void recursion2(int i){
    if( i < 0)
        return;
    recursion2(i - 1);
    System.out.print(i + " ");
}
```
These Methods

Count down from the number entered to 1. The difference between the two however, is what makes recursion special.

recursion1 prints out the current value before calling itself, and recursion 2 does the opposite.

recursion1(5); // outputs:
5 4 3 2 1
and
recursion2(5); // outputs:
1 2 3 4 5
Overloading Methods

Overloading means allowing a method to do different tasks depending on the variables passed.

It's up to the programmer what kinds of parameters will be possible though.

It can be coded for on method to be written any number of times as long as each one has different passes variables.
How can we apply this?

Here are a few methods you can try yourself:

```java
public static int overload(int a){
    return a*5;
}

public static int overload(int a, int b){
    return a*b;
}

public static int overload(char a){
    return (int)a;
}
```
These Methods

All do something different depending on what was passed. What they do in this case is trivial, but it's just important that you understand what's happening.

overload(5); // returns (3 x 5):
15
overload(2, 4); // returns (2 x 4):
8
and
overload('a'); // returns (the ASCII value for 'a'):
97
Nested Loops

Nested loops is term that means loops inside of loops.

This allows a process to keep track of multiple variables that is needs to use in more statements.

Nested loops are applicable in innumerable ways. It is quite common to see them used in sorting algorithms or any other procedures that need to track multiple Objects.
How can we apply this?

Here is a method you can try yourself:

```java
public static double nestedLoop(int dx, int dy, int n){
    double p = 0;
    for(int i = 1; i <= dx; i++)
        for(int j = 1; j <= dy; j++)
            if(i + j == n)
                p++;
    p /= dx*dy*.01;
    return p;
}
```

Can you figure out by looking what it does?
This Method

Calculates the probability of rolling a number (n) given a die with dx sides and another die with dy sides.

```java
nestedLoop(6, 6, 7); // returns: 16.6667
nestedLoop(12, 20, 6); // returns: 2.0833
nestedLoop(4, 6, 4); // returns: 12.5
nestedLoop(1, 1, 2); // returns: 100.0
```