**Understanding Methods:**

**Declaring and calling methods**

**Parameters**

**Exiting a method**

**Returning a value**

**Program command-line arguments**

**Method:**

**Mechanism for organizing code**

**Enables creation of reusable code blocks**

**Can receive data**

**Can return data**

**Name:**

**Same rules and conventions as variable**

**Types parameter list:**

**Allow data values to be passed in**

**Can be empty**

**Name(typed-parameter-list)**

**Body:**

**Consist of zero or more statements**

**Enclosed brackets**

**Return type:**

**Indicates the type of data returned**

**Use void when no value returned**

**Return-type name(typed-parameter-list) {**

**Statements**

**}**

**Method creation (static)**

package com.org.understandingmethods;  
  
public class FirstMethod {  
  
 public static void main(String[] args) {  
 System.*out*.println("Before calling a method");  
 *doSomething*();  
 System.*out*.println("after calling a method");  
 }  
 static void doSomething() {  
 System.*out*.println("Inside method");  
 System.*out*.println("still inside");  
 }  
}

package com.org.understandingmethods;  
  
public class FirstMethod {  
  
 public static void main(String[] args) {  
 System.*out*.println("Before calling a method");  
 *add*(10,20);  
 System.*out*.println("after calling a method");  
 }  
  
 static void add(int a ,int b)  
 {  
 System.*out*.println(a+b);  
 }  
  
}

Method parameters

package com.org.understandingmethods;  
  
public class FirstMethod {  
  
 public static void main(String[] args) {  
 System.*out*.println("Before calling a method");  
 *add*(10, 20, 3);  
 System.*out*.println("after calling a method");  
 }  
/\*  
  
parameters  
 \*/  
 static void add(int a, int b, int count) {  
 int sum = a + b;  
 for (int i = 0; i < count; i++) {  
 System.*out*.println(sum);  
 }  
 }  
  
}

**Method return type**

package com.org.understandingmethods;  
  
public class ReturnValie {  
  
 public static void main(String[] args) {  
 double res = *calculateINterest*(100d, 0.05d, 10);  
 System.*out*.println(res);  
 }  
  
 static double calculateINterest(double amount, double rate, int years) {  
 double interest = amount \* rate \* years;  
 return interest;  
 }  
}

**Cal engine project method & return type implementation**

package com.org.understandingmethods;  
  
public class SwitchStatementArray {  
  
 public static void main(String[] args) {  
  
 double[] leftvals = {100.0d, 25.0d, 22.0d, 11.0d};  
 double[] rightvals = {50.0d, 92.0d, 17.0d, 3.0d};  
 char[] opscode = {'d', 'a', 's', 'm'};  
 double res[] = new double[opscode.length];  
  
 for (int i = 0; i < opscode.length; i++) {  
 res[i] = *execute*(opscode[i], leftvals[i], rightvals[i]);  
 }  
 System.*out*.println(res);  
 for (int i = 0; i < res.length; i++) {  
 System.*out*.println(res[i]);  
 }  
 }  
  
 static double execute(char opCode, double leftvalue, double rightValue) {  
 double result = 0.0d;  
  
 switch (opCode) {  
 case 'a':  
 result = leftvalue + rightValue;  
 break;  
 case 's':  
 result = leftvalue - rightValue;  
 break;  
 case 'm':  
 result = leftvalue \* rightValue;  
 break;  
 case 'd':  
 if (rightValue != 0) {  
 result = leftvalue / rightValue;  
 } else {  
 System.*out*.println("given number val2 is zero");  
 }  
 break;  
 default:  
 System.*out*.println("invalid input");  
 result = 0.0d;  
 }  
 return result;  
 }  
}