Lesson Plan

Methods

PW SKILLS

List of Concepts Involved:

- Methods introduction
- Why are methods important in Java?
- How to declare Methods:
- Call a Method

Topic 1 : Methods in java

- A method is a block of code which only runs when it is called.
- You can pass data, known as parameters, into a method.
- Methods are used to perform certain actions, and they are also known as functions.
- A method is a block of code or collection of statements or a set of code grouped together to perform a certain task or operation.

Topic 2 : Why are methods important in Java?

- A method is a block of code or collection of statements or a set of code grouped together to perform a certain task or operation. We write a method once and use it many times. We are not required to write code again and again.
- Methods in Java are time savers .
- Methods allow us to reuse the code without retyping the code.
- A code created with function is easy to read and dry run. You don't need to type the same statements again and again, instead a function can be called whenever needed.

For example:

Suppose you need to create a program that creates a rectangle and color it. You can solve this problem by creating two methods

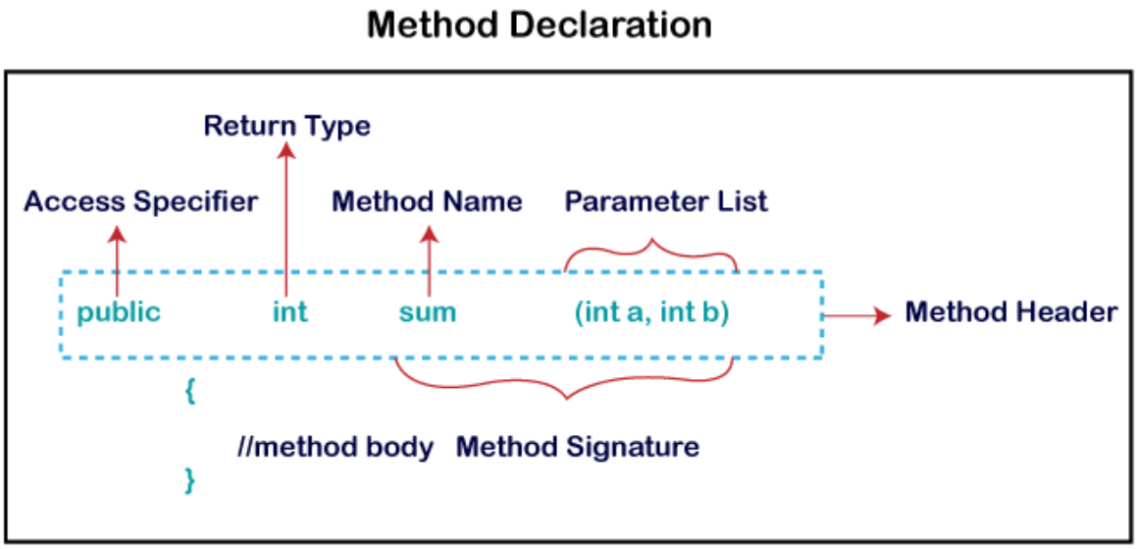
- a method to draw the rectangle
- a method to color the circle

Dividing a complex problem into littler chunks makes your program simple to get it and reusable.In Java, there are two types of methods:

- **User-defined Methods:** You can create your own method based on your requirements.
- **Standard Library Methods:** These are built-in methods in Java that are available to use.

Topic 3: How to declare Methods:

The method declaration provides information about the attributes, such as visibility, return-type, name, and arguments of methods. It consists mainly of six components that are known as method header, as shown in the following figure.



1. **Method Signature:** In Java, a method signature is part of the method declaration. It's the combination of the method name and the parameter list.

2. **Access Specifier:** Access specifier is the access type of the method. It specifies the visibility of the method. In java, there are four types of access specifier:

i) Private : The methods or data members declared as private are accessible only within the class in which they are declared.

ii) Public : Methods that are declared as public are accessible from everywhere in the program. There is no restriction on the scope of public data members.

iii) Protected : The methods declared as protected are accessible within the same package or subclasses in different packages.

iv) Default : When no access modifier is specified for a method – It is said to be having the default access modifier by default. Default access modifiers are accessible only within the same package.

3. **Return type :** It defines and constrains the data type of the value returned from a method.

4. **Method Name :** Each method has its own name and method is invoked by its name.Name should be given in such a way that it corresponds to the functionality of the method.

5. **Parameters :** Parameters act as variables inside the method. Parameters are specified after the method name, inside the parentheses. You can add as many parameters as you want, just separate them with a comma.

6. **Method Body:** It is a part of the method declaration. It contains all the actions to be performed. It is enclosed within the pair of curly braces.

Topic 4 : Call a Method

- To call a method in Java, you have to write the method's name followed by two parentheses () and a semicolon;.
- To call the method you have to know the name of the method that we are calling and number of parameters required and their data types and the return type of the method to collect the returned value by the method.

In the following example, welcome() method is used to print a text (the action), when it is called:

Example 1

```
public class Main {
  static  void welcome() {
    System.out.println("Welcome to Physics Wallah");
  }

  public static void main(String[] args) {
    welcome();
  }
}
```

Output: Welcome to Physics Wallah

Explanation:

In the above example, we have created a method named welcome(). The method takes no parameters. In the line,
welcome();
Here, we have called the method welcome() by passing no arguments . Since the method is not returning any value, it's return type is void.

Example 2. Java Program to add two numbers using methods

```
class Main {

  // create a method
  public int addition(int p, int q) {
    int add = p + q;
    // return value
    return add;
  }
```

```
  public static void main(String[] args) {

    int p = 40;
    int q = 60;

    // create an object of Main
    Main obj = new Main();
    // calling method
    int answer = obj.addition(p,q);
    System.out.println("Sum is: " + answer);
  }
}
```

Output: Sum is 100

Explanation:

In the above example, we have created a method named addition(). The method takes two parameters p and q. In the line,
int answer = obj.addition(p, q);
Here, we have called the method by passing two arguments p and q. Since the method is returning integer value, we have stored the value in the answer variable of int type.

Note- If the method does not return any value, we use the void keyword as the return type of the method.

Example 3.

```
 class Main {

  // create a method
  public void addition(int p, int q) {
    int add = p + q;
    // print value
    System.out.println("Sum is: " + add);
  }

  public static void main(String[] args) {

    int p = 40;
    int q = 60;

    // create an object of Main
    Main obj = new Main();
    // calling method
    obj.addition(p,q);

  }
}
```

Output: Sum is 100

Explanation:

In the above example, we have created a method named addition(). The method takes two parameters p and q. In the line,
obj.addition(p, q);
Here, we have called the method by passing two arguments p and q. Since the addition method return type is void so it is not returning any value.

Standard Library Methods

The standard library methods are built-in methods in Java that are readily available for use. These standard libraries come along with the Java Class Library (JCL) in a Java archive (*.jar) file with JVM and JRE.

For example,

- print() is a method of java.io.PrintSteam. The print("...") method prints the string inside quotation marks.
- sqrt() is a method of Math class. It returns the square root of a number.
- floor() is a method of Math class. It returns the rounded-off nearest integer which is less than the given value.

Here's a working example:

Example :

```
public class Main {
  public static void main(String[] args) {
    // using the sqrt() method
    System.out.print("Square root of 16 is: " +
Math.sqrt(16));
    }
}
```

Output: Square root of 16 is: 4.0