

Java - Introduction to Programming

Lecture 7

Methods/Functions

A function is a block of code that performs a specific task.

Why are functions used?

- If some functionality is performed at multiple places in software, then rather than writing the same code, again and again, we create a function and call it everywhere. This helps reduce code redundancy.
- Functions make maintenance of code easy as we have to change at one place if we make future changes to the functionality.
- Functions make the code more readable and easy to understand.

The **syntax** for function declaration is :

```
return-type function_name (parameter 1, parameter2, ..... parameter n){  
    //function_body  
}  
return-type
```

The **return type** of a function is the data type of the variable that that function returns.

For eg - If we write a function that adds 2 integers and returns their sum then the return type of this function will be 'int' as we will return a sum that is an integer value.

When a function does not return any value, in that case the return type of the function is 'void'.

function_name

It is the unique name of that function.

It is always recommended to declare a function before it is used.

Parameters

A function can take some parameters as inputs. These parameters are specified along with their data types.

For eg- if we are writing a function to add 2 integers, the parameters would be passed like –

```
int add (int num1, int num2)
```

main function

The main function is a special function as the computer starts running the code from the beginning of the main function. Main function serves as the entry point for the program.

Example :

```
package com.apnacollege;

public class Main {
    //A METHOD to calculate sum of 2 numbers - a & b
    public static void sum(int a, int b) {
        int sum = a + b;
        System.out.println(sum);
    }

    public static void main(String[] args) {
        int a = 10;
        int b = 20;
        sum(a, b); // Function Call
    }
}
```

Qs. Write a function to multiply 2 numbers.

```
import java.util.*;

public class Functions {

    //Multiply 2 numbers

    public static int multiply(int a, int b) {

        return a*b;

    }

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);
```

```

        int a = sc.nextInt();

        int b = sc.nextInt();

        int result = multiply(a, b);

        System.out.println(result);

    }
}

```

Qs. Write a function to calculate the factorial of a number.

```

import java.util.*;

public class Functions {

    // public static int calculateSum(int a, int b) {
    //     int sum = a + b;
    //     return sum;
    // }

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

    public static void printFactorial(int n) {
        //loop
        if(n < 0) {
            System.out.println("Invalid Number");
            return;
        }
        int factorial = 1;

        for(int i=n; i>=1; i--) {
            factorial = factorial * i;
        }

        System.out.println(factorial);
        return;
    }
}

```

```
public static void main(String args[]) {  
    Scanner sc = new Scanner(System.in);  
    int n = sc.nextInt();  
  
    printFactorial(n);  
}  
}
```

Qs. Write a function to calculate the product of 2 numbers.

```
import java.util.*;  
  
public class Functions {  
  
    // public static int calculateSum(int a, int b) {  
  
        // int sum = a + b;  
  
        // return sum;  
  
    // }  
  
    public static int calculateProduct(int a, int b) {  
  
        return a * b;  
  
    }  
  
    public static void main(String args[]) {  
  
        Scanner sc = new Scanner(System.in);  
  
        int a = sc.nextInt();  
  
        int b = sc.nextInt();  
  
        System.out.println(calculateProduct(a, b));  
  
    }  
}
```

Homework Problems

1. Make a function to check if a number is prime or not.
2. Make a function to check if a given number n is even or not.
3. Make a function to print the table of a given number n .
4. Read about Recursion.