

Day-22

11 July 2023 15:06

1. METHOD NOT ACCEPTING PARAMETER AND NOT RETURNING VALUE
2. METHOD NOT ACCEPTING PARAMETERS BUT RETURNING VALUE
3. METHOD ACCEPTING PARAMETER LIST BUT NOT RETURNING VALUE
4. METHOD ACCEPTING PARAMETER LIST AND RETURNING VALUE

METHOD NOT ACCEPTING PARAMETER AND NOT RETURNING VALUE

```
import java.util.Scanner;

class Test1
{
    public static void main(String[] args)
    {
        add();
    }

    public static void add()
    {
        int a = 1, b = 2;
        int sum = a+b;

        System.out.println(sum);
    }
}
```

METHOD NOT ACCEPTING PARAMETERS BUT RETURNING VALUE

```
import java.util.Scanner;

class Test1
{
    public static void main(String[] args)
    {
        int result = add();
        System.out.println(result);
    }
}
```

```

        public static int add()
        {
            int a = 10, b = 20, c=30;
            int sum = a+b+c;

            return sum;

        }

    }

```

METHOD ACCEPTING PARAMETER LIST BUT NOT RETURNING VALUE

```

import java.util.Scanner;

class Test1
{
    public static void main(String[] args)
    {
        add(10,20,30);
    }

    public static void add(int a, int b, int c)
    {
        int sum = a+b+c;

        System.out.println(sum);

    }

}

```

METHOD ACCEPTING PARAMETER LIST AND RETURNING VALUE

```

import java.util.Scanner;

class Test1
{
    public static void main(String[] args)
    {
        int result=add(10,20,30);
    }
}

```

```

        System.out.println(result);
    }

    public static int add(int a, int b, int c)
    {

        int sum = a+b+c;
        return sum;

    }

}

```

PRACTICE QUESTIONS

TASK-1

1. WRITE A PROGRAM ON METHOD ACCEPTING 2 INTEGER VALUES AND RETURNS A BOOLEAN VALUE
2. WRITE A PROGRAM ON METHOD ACCEPTS 1 DOUBLE VALUE AND 1 INTEGER VALUE AND RETURNS A CHAR VALUE
3. WRITE A PROGRAM ON METHOD ACCEPTS STRING VALUE AND RETURNING STRING VALUE
4. WRITE A PROGRAM ON METHOD ACCEPTING INTEGER VALUE , DOUBLE , STRING AND RETURNS A BOOLEAN VALUE

TASK-2

1. WRITE A PROGRAM TO FIND THE NUMBER IS EVEN OR ODD USING METHODS AND YOU SHOULD READ THE NUMBERS FROM THE USER
2. WRITE A PROGRAM TO CALCULATE THE % OF STUDENTS MARKS USING METHODS , READ 4 MARKS VALUES FROM THE USER
3. WRITE A PROGRAM TO FIND THE GIVEN NUMBER IS PALINDROME OR NOT USING METHODS AND READ THE NUMBER FROM THE USER
4. WRITE A PROGRAM TO FIND THE NUMBER IS SPECIAL NUMBER OR NOT USING METHODS AND READ NUMBER FROM THE USER
5. WRITE A PROGRAM TO FIND THE GREATEST OF 3 NUMBERS USING METHODS AND READ THE NUMBERS FROM THE USER
6. WRITE A PROGRAM TO FIND THE NUMBER IS ARMSTRONG NUMBER OR NOT USING METHODS AND READ THE NUMBER FROM THE USER

IMPORTANT CONCLUSIONS OF METHODS

1. THE PROCESS OF INVITING A METHOD OF IMPLEMENTATION IS CALLED METHOD CALLING/INVOKING
2. IF WE ARE DEFINING A METHOD AND NOT CALLING FOR IMPLEMENTATION THERE WILL NOT BE ANY COMPILATION ERROR BUT IF WE ARE CALLING A METHOD AND NOT DEFENING THAT METHOD THEN WE WILL GET COMPILATION ERROR

```
class Test1
```

```

{
    public static void main(String[] args)
    {
        m1();
    }
}

```

3. A METHOD CAN ACCEPT ANY NUMBER OF VALUES BUT IT CAN RETURN ONLY ONE VALUE

```

class Test1
{
    public static void main(String[] args)
    {
        int a = m1(1, 1.1, 'c', "BitsQ");
        System.out.println(a);
    }

    public static int m1(int a, double b, char c, String d)
    {
        return a;
    }
}

```

4. IT IS NOT MANDATORY FOR THE TYPE OF VALUE WHICH WE ARE PASSING (PARAMETER LIST) TO BE SAME AS RETURN TYPE OF METHOD.
5. THE RETURN STATEMENT SHOULD ALWAYS BE LAST STATEMENTS OF METHOD BODY, IF AT ALL WE WRITE ANY STATEMENT AFTER RETURN STATEMENT WE WILL GET COMPILATION ERROR

EX :

```

class Test1
{
    public static void main(String[] args)
    {
        int a = m1(1, 1.1, 'c', "BitsQ");
        System.out.println(a);
    }

    public static int m1(int a, double b, char c, String d)
    {
        return a;
        System.out.println("HI"); //unreachable statement error
    }
}

```

```
}
```

6. A METHOD CAN HAVE MULTIPLE RETURN STATEMENT IF WE ARE WORKING WITH ANY BRANCHING STATEMENT BUT IF WE WRITE MULTIPLE RETURN STATEMENTS INDEPENDENTLY WE GET COMPILLATION ERROR

```
class Test1
{
    public static void main(String[] args)
    {
        m1();
    }

    public static boolean m1()
    {
        return true;
        return false;
    }
}
```