Working with Methods:

"A function | method is self-contained block, which is having a statement or group of statement to perform a particular task"

Methods are classified into 2 types

- Predefined methods
  - Methods which are provided by java language compiler is called predefine methods also known as built-in methods
- User defined methods
  - The methods which are declared by us, Based on our programming requirements

Methods Prototypes:

- These are divided into following types based on their arguments and their return values
  - Method without argument and without return value
  - Method without argument , with return value
  - Method with argument ,without return value
  - Method with argument , with return value

Syn For method Definition:

```
[modifiers]  <return type>  <methodName>([list of args])
{
   [local variables];
  Statement(s);
  [return <value | variable |exp>];
}
```

<u>Formal parameter vs actual arguments:</u>

- The variables which are declared in the function definition are called formal parameters
- Formal parameter must be variables with corresponding datatypes
- Formal parameter are copies of actual arguments
- Formal parameter acts as local variables

<u>Actual Arguments:</u>

- The Parameters which we are passing by the time of calling the functions
- Actual arguments can be values , variables, expressions

Note: Type of both actual and formal parameter must be same, but they are not in the same location

```
1. class Demo
2. {
3.         void greetings()  //defining mtd
4.         { System.out.println("Hello MyDear ");
5.            System.out.println("Have a nice Day ...");
6.         }
7.
8.         public static void main(String args[ ])
9.         {
10.                    Demo d=new Demo();
11.                           d.greetings(); //Calling mtd
12.                                  d.greetings();
13.                                  d.greetings();
14.                                  d.greetings();
15.              }
16.        }
17.
18.        Ex2:
19.        class Demo2
20.        {
21.              void sum(int x,int y)
22.              {   int s;
23.                   s=x+y;
24.                   System.out.println("Sum is : "+s); }
25.
26.              public static void main(String args[ ])
27.              {
28.               Demo2 d=new Demo2();
29.                                d.sum(10,20);   //Calling Mtd
30.                                          d.sum(5+5+5,10+10+10);
31.                         int a=90,b=90;   //local variable
```

```
                                        d.sum(a,b);
        }
}

Ex3:
class Test
{
        int sq(int x)
        {    int s;
                s=x*x;
                return s;
        }

        public static void main(String args[ ])
        {
         Test t=new Test();
                        int r=t.sq(9);
                 System.out.println("Result is :::> "+r);

        }
}

Ex 4:
class Circle
{  //No instance fields

        float findArea(float rad)
        {   float area;
            area=(3.14f * rad *rad);  //area=12.5600
            return area; }

    float findCF(float rad)
        {    float c;
              c=(2*3.14f*rad);
              return c;}
```

```
67.
68.             public static void main(String args[])
69.             {
70.                     Circle c=new Circle( );
71.                          float a=c.findArea(3.0f);
72.                     System.out.println("Result is : "+a);
73.
74.                          float cc=c.findCF(3.0f);
75.                                  System.out.println("CF Circle is : "+cc);
76.             }
77.     }
78.
```

Imp Note:

- Whenever you want use the value calculated by a function outside of that function then we have return the value
- Whenever a function or methods is not returning any value to the calling function then we have write “void” as return type