functions

function is a tool in programming that helps you refactor / segregate the code.

Problem :- * Take salary of n employees in input

* Add 10% increment
* find the average salary of all emps.
* Print individual emp salary after inc.

```
psv main() {
    scanner  ___  _            3 emps.
    sal1 , sal2 , sal3  → IP
    newsal1 , new_sal2, new_sal3  → Increment
    avg
    print
}
```

A function contains a piece of code specific to a task

syntax :

* function calling
* function definition ✓ : will always return a value

```
psv main() {
    scanner ----
    ----          }  Input
    ----
    INC           }  Increment
```

AVG] Average

Print salaries of each emp] Print

}

Function definition

```
salaryInc ( ) {
    // 50 lines
    contains the code responsible for salary increment

}
```

Function def^{n}

```
findsalAvg ( ) {
    // 50 lines
    contains the code responsible for finding average
}
```

fun^{n} def^{n}

```
printsalaries ( ) {
    // 50 lines

}
```

```
psv main ( ) {
```

Input

```
    Scanner ______
    __ __ ____ _

    ___ ___ __
```

salaryInc()
salaryInc()
salaryInc()

salaryInc ( ); → function call

findsalaryAvg ( ); → function call

printsalaries ( ); → function call

```
}
```

Function definition :

```
returnType functionName ( parameters/arguments) {



}
```

Parameter is extra variables that are used in f^n def^n.

int x
char ch
bool var
String str

returnType is the type of value that function returns

void	:	f^n	def^n	returns	nothing
int	:	f^n	def	returns	integer
char	:	"	"	"	character
bool	:	"	"	"	boolean
string.	:	"	"	"	string

* A function definition either returns integer , charcter ,string etc or nothing.

→ argument

```
int salaryInc ( int sal) {
        int newsal = sal + sal/10 ;
        return newsal;
}

psv main ( ) {

        Scanner ---
        -- -
        ____

        salaryInc( 100);
}
```

parameter

→ sal : 10

: only 1 parameter tha is integer

* salaryInc function will have int returnTyp since it will return the new incremented salary. because this f^n has to tell main f^n that the new inc sallary is 110.

* Main f^n has to pass the original salary in parameter to tell salaryInc f^n

Working Example

```
public class Main {

    static int salaryIn(int sal){
            int newsal = sal + sal/10;
            return newsal;
    }

    public static void main(string [] args){
            scanner sc= new scanner(S4--);
            int sal= sc.nextInt();

            int incsal= salaryInc(sal);
            SOUT(incsal);
    }
```

function definition of salaryInc function

calling salaryInc function ←