Definitions:

1. Function: Function is a block of program code that performs a particular task. It is embedded in the main program and can be called when needed to perform the specified task. A function also works like a data processing module in which data is given as input arguments which processed as defined inside and finally the result is prepared to return in the calling section.
There are two types of functions:-
A. Library function:- these are in-built function stored in Java class library. When we want to use a function, its concerned class is imported in the present class. Use of library function makes writing a program easy.
B. User defined function:- it is written by the programmer to perform specific task. It is written out of main function and inside the class.

2. Function definition and declaration: When the prototype of the function is written with specification of function name, its input variable name with type and data type to return, it is called function declaration. Actual task to perform within the function is written using different commands within a pair of braces, along with function prototype specified before this. This is called function definition.

3. Argument or input argument: The values given from the calling section of the program to the function as data items to be manipulated are called input arguments. These work like input entities for a processing module. Soon after the function name, it is given within a pair of parenthesis, having values separated by comma. The number of items and its sequence must be same as specified in the function declaration or its prototype.

4. Return value of a function: When the function generates a value to be sent in the calling section of the program it is called the Return value of a function. Generally it specified with return() keyword at the end of function module. When such function is called in main section of the program, function along with its arguments is used to assign a variable that holds the return value.

Syntax for defining a function:

```
<static keyword><return type><function name>(argument1,argument2…...){
statement(s);
}
```

static is a keyword written before the function name.
return type is the type of data returned by function.
function name is the name by which it is called in main function.
arguments are given as data type followed by identifier name. It used to pass parameters from the calling section.
statement(s) are written within a pair of curly braces. It is consisting of single statement or block of statements to perform a task.

Advantages of using function:-
1. It makes a complex task broken in modules.
2. Debugging is easy.
3. Modification is easy.
4. A function can be called many times with different set of arguments.

5. Array: Array is an object that works like a data container holding more than one value with one name. It is similar like indexed variable like X_1, X_2, X_3….. used in mathematical calculations, here X_i represents a arbitrary i^{th} item.

6. Array Index: Different values stored in an array are addressed by their specific count using whole numbers. This count is called Array Index. The count starts from zero that is, first value of an array is referred by specifying 0 index with array name.

variable	ar[0]	ar[1]	ar[2]	ar[3]	ar[4]	ar[5]	ar[6]
index	0	1	2	3	4	5	6
Value	45	50	23	36	47	19	97

Syntax for array declaration:

<data type>[number of elements] <array name>

eg. int[5] ar;

assigning value in array:

int ar[7]={45,50,23,36,47,19,97};

or

int[] ar=new ar[7];

ar[0]=45;ar[1]=50;ar[2]=23;ar[3]=36;ar[4]=47;ar[5]=19];ar[6]=97;

Program to calculate sum of two entered numbers using function.

Ans: The given task demands to define a function for sum that should have input arguments as well as the return type. In the main section, two numbers are raised, after being assigned by the user at run time. After calculating final value, it is returned in the calling section. (input argument and return value in main section)

```
import java.util.Scanner;
public class calc{
	public static void main(String[] args){
		int num1,num2,s;
		Scanner x=new Scanner(System.in);
		System.out.println("Enter to numbers to calculate sum");
		num1=x.nextInt();
		num2=x.nextInt();

		s=add(num1,num2);
		System.out.println("sum of two numbers ="+s);
	}

	static int add(int n1,int n2){
		int sum;
		sum=n1+n2;
		return sum;
	}
}
```

Program to print table of entered number.

Note:-

In this program user return type of the function is void(empty) so it does not return anything in the calling section. In main section, function is called with its name and an

argument of same data type so it only performs the task specified within. (input argument in main section, but showing result in the function itself)

```
import java.util.Scanner;
public class MulTable{
	public static void main(String args[]){
		int num;
		Scanner x=new Scanner(System.in);
		System.out.println("Enter to numbers to print table");
		num=x.nextInt();
		table(num);
	}
	static void table(int t){
		int i;
		for(i=1;i<=10;i++){
			System.out.println(t*i);
		}
	}
}
```

Program to calculate area of a rectangle with entered values.
Note:-
In this program, the function has no input arguments, no return type. The function is called only by its name. After being called, function itself assigns input, calculates given task and shows the result. (No input argument, no return in main. Function itself assigns values, calculates result and shows it)

```
import java.util.Scanner;
public class RectArea{
	public static void main(String args[]){
		area();
	}

	static void area(){
		float length,breadth,a;
		Scanner x=new Scanner(System.in);
		System.out.println("Enter length and breadth to calculate area of a
rectangle");
		length=x.nextFloat();
		breadth=x.nextFloat();
		a=length*breadth;
		System.out.println("Area of a rectangle="+a);
	}
}
```

Program to print values of array in forward and reverse order.

```
public class NumAr{
	public static void main(String args[]){
		int[] num=new int[10];

		num[0]=100;
		num[1]=120;
		num[2]=90;
		num[3]=115;
		num[4]=53;
		num[5]=20;
		num[6]=70;
		num[8]=14;
		num[9]=160;

		for(int i=0;i<10;i++){
			System.out.println(num[i]);
		}

		for(int i=9;i>=0;i--){
			System.out.println(num[i]);
		}

	}
}
```

Program to calculate average of alternate numbers of an array.

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

		int[] num={10,12,15,3,20,7,14,16,18,5};
		float sum=0,avg=0;
		for(int i=0;i<10;i+=2){
			sum+=num[i];
		}
		avg=sum/10;
		System.out.println("Average of alternate values of array="+sum);
	}
}
```