```java

public class EasyFunctionsCode{

    // Write a method in Java that returns "Hello" and takes no input.
    public static String GetHello(){
        return "Hello";
    }
     // Declare a method that takes a String name and returns a greeting.
     public static String GetGreeting(String name){
        return "Namaste "+ name;
    }
    // Write a method that takes two integers and returns their sum.
    public static int getSum(int num1,int num2){
        return num1+num2;
    }
    // Create a method that returns the square of an integer.
    public static int getSquare(int x){
        return x*x;
    }
    // Write a method that returns area of a rectangle given width and height.
    public static int getRectArea(int len,int breadth ){
        return len*breadth;
    }

     // Declare a method that accepts three floats and returns their average.
     public static double getAveOf3( double num1, double num2, double num3){
        return (num1+num2+num3)/3;
     }

```

```java
    // Create a method that takes an array of integers and returns their sum.
    public static int getArraySum(int[] nums){
        int sum=0;
        for(int num : nums){
            sum+=num;
        }
        return sum;
    }
    // Write a method that checks if a number is even.
    public void evenOrOdd(int num){
        String result = ((num&1)==1)?"Odd NUmber":"Even Number";
        System.out.println("The Number "+num+" is a "+result);
    }
    // Define a method that returns a string repeated n times.
    public static String getRepeatedStriing(String s,int n){
        StringBuilder sb=new StringBuilder();
        for(int i=0;i<n;i++){
            sb.append(s);
        }
        return sb.toString();
    }
```

```
4   public class EasyFunctionsCode{
53      //Write a method merges two arrays
54      public static int[] mergeTwoArray(int [] arr1,int []arr2){
55      int p=0;
56      int mergedArr[]=new int[arr1.length+arr2.length];
57      for(int i=0;i<arr1.length;i++){
58          mergedArr[p++]=arr1[i];
59      }
60      for(int i=0;i<arr2.length;i++){
61          mergedArr[p++]=arr2[i];
62      }
63      return mergedArr;
64      }
65      // Write a method that returns the maximum element from an array.
66      public static int findMaximumNumArray(int[] nums){
67          int max=Integer.MIN_VALUE;
68          for(int num:nums){
69              if(num>max)max=num;
70          }
71          return max;
72      }
```

```java
    // Write a method to reverse a string.
    public static String reverse(String s){
        StringBuilder sb = new StringBuilder(s);
        int right=s.length()-1;
        int left=0;
        while (left < right) {

            char temp = sb.charAt(left);
            sb.setCharAt(left, sb.charAt(right));
            sb.setCharAt(right, temp);
            left++;
            right--;
        }
        return sb.toString();

    }
    // Create a method that checks if a string is a palindrome.

    public static boolean isPalindrom(String s){
        if(s.equals(reverse(s)))return true;
        return false;
    }

    // Write a method to calculate factorial of a number.
    static Map<Integer,Integer>factMap=new HashMap<>();
    public static int findFact(int n){
        if(n==0||n==1)return 1;
        if(factMap.containsKey(n))return factMap.get(n);
        int fact = n*findFact(n-1);
        factMap.put(n,fact);
        return fact;
    }
```

```java
// Define a method that returns the nth Fibonacci number.

public static int findFib(int n){
    if (n == 0) return 0;
    if (n == 1 || n == 2) return 1;
    int [] dp = new int[n+1];
    dp[0]=0;
    dp[1]=1;
    dp[2]=1;
    for(int i=3;i<dp.length;i++){
        dp[i]=dp[i-1]+dp[i-2];
    }
    return dp[n];
}
```

Run | Debug | Run main | Debug main