

Explanation of sumoftwonumber code

1. **Import Scanner class** At the very top, import java.util.Scanner; brings in the Scanner tool.  
    This tool helps the program read what the user types on the keyboard.
2. **Define the main class and method** The program is inside a class named sumoftwo.  
    The main part where the program starts running is the main method:  
    public static void main(String[] args)
3. **Create a Scanner object to take input** Scanner sc = new Scanner(System.in);  
    This line creates a Scanner called sc that listens to the keyboard input.
4. **Ask the user to enter the first number** System.out.print("Enter first number: ");  
    This prints a message on the screen asking the user for the first number.
5. **Read the first number from the user** int a = sc.nextInt();  
    The program waits for the user to type an integer and presses Enter, then stores that number in variable a.
6. **Ask the user for the second number** System.out.print("Enter second number: ");  
    This asks the user to enter the second number.
7. **Read the second number** int b = sc.nextInt();  
    The second number typed by the user is stored in variable b.
8. **Calculate the sum of the two numbers** int sum = a + b;  
    This adds the two numbers a and b and saves the result in sum.
9. **Display the result** System.out.println("Sum: " + sum);  
    The program prints out the sum with the message "Sum:".
10. **Close the Scanner** sc.close();  
     This closes the Scanner to free up resources. It's good practice but not mandatory in small programs.

Explanation of reversestring code

### **Step-by-step explanation:**

1. **Import Scanner class** import java.util.Scanner; lets the program use Scanner to read what the user types on the keyboard.
2. **Define the main class and method** The program is inside a class named reversestring.  
    The program starts running in the main method:  
    public static void main(String[] args)
3. **Create a Scanner object** Scanner sc = new Scanner(System.in);  
    This sets up the Scanner named sc to listen for keyboard input.
4. **Prompt the user to enter a string** System.out.print("Enter a string to reverse: ");  
    This message asks the user to type any string.
5. **Read the whole line typed by the user** String input = sc.nextLine();  
    This stores the entire string the user typed into the variable input.
6. **Initialize an empty string to hold the reversed string** String reversed = "";  
    This will gradually build up the reversed version of the original string.

**Reverse the string using a loop** java  
Copy code  
for (int i = input.length() - 1; i >= 0; i--) {

reversed += input.charAt(i);

}

* + The loop starts from the last character of the input (input.length() - 1) and moves backwards to the first character (index 0).
  + Each character is added to the reversed string one by one.
  + So, if the input was "hello", the loop takes 'o', then 'l', then 'l', 'e', 'h', building the reversed string "olleh".

1. **Print the reversed string** System.out.println("Reversed string: " + reversed);  
    This displays the reversed string to the user.
2. **Close the Scanner** sc.close();  
    This closes the Scanner object to free resources.

Explanation of prime number code

1. **Import Scanner class** import java.util.Scanner; lets the program take input from the user through the keyboard.
2. **Define the main class and method** The program is inside a class named primenumber.  
    The program starts running inside the main method.
3. **Create a Scanner object** Scanner sc = new Scanner(System.in);  
    This creates a Scanner named sc to read user input.
4. **Prompt the user to enter a number** System.out.print("Enter a number to check if it is prime: ");  
    This asks the user to type a number.
5. **Read the number from the user** int num = sc.nextInt();  
    The number typed by the user is stored in the variable num.
6. **Initialize a boolean variable to track if the number is prime** boolean isPrime = true;  
    We assume the number *is* prime at first until proven otherwise.

**Check if the number is less than or equal to 1** java  
Copy code  
if (num <= 1) {

isPrime = false;

}

1. Numbers 1 and below are **not** prime, so we mark isPrime as false immediately.

**Check divisibility from 2 up to the square root of the number** java  
Copy code  
else {

for (int i = 2; i <= Math.sqrt(num); i++) {

if (num % i == 0) {

isPrime = false;

break;

}

}

}

* + We try dividing the number by every integer i starting from 2 up to √num.
  + If num is divisible by any i (meaning num % i == 0), it's **not** prime.
  + In that case, we set isPrime to false and stop checking further (break).

**Print the result** java  
Copy code  
if (isPrime) {

System.out.println(num + " is a prime number.");

} else {

System.out.println(num + " is not a prime number.");

}

1. This shows whether the number is prime or not.
2. **Close the Scanner** sc.close(); frees up resources by closing the Scanner.

Largest among three explanation

1. **Import Scanner class** import java.util.Scanner;  
    This lets the program take input from the user.
2. **Define the main class and method** The program is inside the class named largeramongthree.  
    The program starts running inside the main method.
3. **Create a Scanner object** Scanner sc = new Scanner(System.in);  
    This sets up the Scanner to read what the user types.
4. **Ask the user to enter the first number** System.out.print("Enter first number: ");  
    Then store the entered number in variable a:  
    int a = sc.nextInt();
5. **Ask the user to enter the second number** Similarly, print the message and read input into variable b.
6. **Ask the user to enter the third number** Print the message and read input into variable c.
7. **Determine the largest number** Use if-else statements to compare the three numbers:  
   * If a is greater than or equal to both b and c, then a is the largest.
   * Else if b is greater than or equal to both a and c, then b is the largest.
   * Otherwise, c is the largest.
8. **Print the largest number** Display the largest number to the user with a message.
9. **Close the Scanner** sc.close(); to free resources.

Code explanation of find factorial

### **What is factorial?**

* Factorial of a number n (written as n!) means multiplying all whole numbers from 1 up to n.
* For example, 5! = 5 × 4 × 3 × 2 × 1 = 120.

### **Step-by-step explanation of the code:**

1. **Import Scanner class** import java.util.Scanner; lets the program get input from the user.
2. **Define the main class and method** The program is inside the class called findfactorial.  
    The program starts running in the main method.
3. **Create a Scanner object** Scanner sc = new Scanner(System.in);  
    This sets up the Scanner to read the number you type.
4. **Ask the user to enter a number** System.out.print("Enter a number to find its factorial: ");  
    This message prompts the user.
5. **Read the input number** int num = sc.nextInt();  
    This stores the number entered by the user in the variable num.
6. **Initialize a variable to store the factorial** long factorial = 1;  
    We start with 1 because multiplying by 1 doesn’t change the result.

**Calculate the factorial using a loop** java  
Copy code  
for (int i = 1; i <= num; i++) {

factorial \*= i;

}

* + The loop runs from 1 up to num.
  + Each time, it multiplies the current factorial value by i.
  + So if num is 5, the factorial calculation will do: 1 × 1, then × 2, then × 3, × 4, and finally × 5.

1. **Print the factorial result** System.out.println("Factorial of " + num + " is: " + factorial);  
    This displays the calculated factorial.
2. **Close the Scanner** sc.close(); releases resources.

Evenorodd code explanation

**Import Scanner class** import java.util.Scanner; lets the program take input from the keyboard.

**Define the main class and method** The program is inside a class named evenorodd.  
 It starts running inside the main method.

**Create a Scanner object** Scanner sc = new Scanner(System.in);  
 This sets up the Scanner to listen for user input.

**Ask the user to enter a number** System.out.print("Enter a number: ");  
 This prints a message asking the user to type a number.

**Read the number from the user** int num = sc.nextInt();  
 The program stores the entered number in the variable num.

**Check if the number is even or odd** java  
Copy code  
if (num % 2 == 0) {

System.out.println(num + " is even.");

} else {

System.out.println(num + " is odd.");

}

* The % operator calculates the remainder when num is divided by 2.
* If the remainder is 0 (num % 2 == 0), the number is even.
* Otherwise, it is odd.

**Close the Scanner** sc.close(); frees resources by closing the Scanner.