

Question1). Write a java program Add two Numbers

Solution:-import java.util.Scanner;

```
public class AddTwoNumbers {  
  
    public static void main(String[] args) {  
  
        // Create a Scanner object to read input from the user  
  
        Scanner scanner = new Scanner(System.in);  
  
  
        // Prompt the user to enter the first number  
  
        System.out.print("Enter the first number: ");  
  
        int number1 = scanner.nextInt();  
  
  
  
        // Prompt the user to enter the second number  
  
        System.out.print("Enter the second number: ");  
  
        int number2 = scanner.nextInt();  
  
  
  
        // Add the two numbers  
  
        int sum = number1 + number2;  
  
  
  
        // Display the result  
  
        System.out.println("The sum of the two numbers is: " + sum);  
  
  
  
        // Close the scanner  
  
        scanner.close();  
  
    }  
}
```

Explanation :-

1. The program starts by importing the Scanner class from the java.util package. This class allows us to read input from the user.
2. The public class AddTwoNumbers line defines a class named AddTwoNumbers, which contains the main method where the program execution begins.
3. Inside the main method, we create a Scanner object named scanner to read input from the user.
4. We prompt the user to enter the first number by displaying the message "Enter the first number: " using the System.out.print method.
5. The program waits for the user to enter a number, and then we use the nextInt method of the Scanner object to read that number and store it in the variable number1.
6. Similarly, we prompt the user to enter the second number and store it in the variable number2.
7. We add the two numbers together using the + operator and store the result in the variable sum.
8. Finally, we display the result to the user by printing the message "The sum of the two numbers is: " followed by the value of the sum variable using the System.out.println method.
9. We close the Scanner object to release any system resources associated with it.
10. The program prompts the user for two numbers, adds them together, and then displays the sum.

QUESTION 2). Write a java program Check Whether a Number is Even or Odd,

```
SOLUTION:- public class EvenOddChecker {  
  
    public static void main(String[] args) {  
  
        int number = 17;  
  
        if (number % 2 == 0) {  
  
            System.out.println(number + " is an even number.");  
  
        } else {  
  
            System.out.println(number + " is an odd number.");  
  
        }  
  
    }  
  
}
```

EXPLANATION;-

1. We start by declaring a class named EvenOddChecker. The public keyword indicates that the class is accessible from other classes. The class contains a single method named main.
2. Inside the main method, we declare an integer variable named number and assign it a value of 17. This is the number we want to check.
3. We use an if statement to check whether number is divisible by 2 without a remainder. The % operator is the modulo operator, which returns the remainder of the division. If number % 2 equals 0, it means number is divisible by 2 and therefore an even number.
4. If the condition in the if statement is true, we print a message indicating that the number is even using System.out.println().
5. If the condition in the if statement is false, it means the number is not divisible by 2 and therefore an odd number. In this case, we print a message indicating that the number is odd using System.out.println().
6. After executing the appropriate println statement, the program ends.

QUESTION3). Write a java program Check if a given number is palindrome or not

SOLUTION;- public class CheckPalindrome {

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

```
        int number = 12321;
```

```
        int originalNumber = number;
```

```
        int reversedNumber = 0;
```

```
        while (number != 0) {
```

```
            int digit = number % 10;
```

```
            reversedNumber = reversedNumber * 10 + digit;
```

```
            number /= 10;
```

```
        }
```

```
        if (originalNumber == reversedNumber) {
```

```
            System.out.println(originalNumber + " is a palindrome.");
```

```
        } else {
```

```

        System.out.println(originalNumber + " is not a palindrome.");
    }

}

}

```

EXPLANATION;-

1. In this program, we have assigned the value 12321 to the variable number.
2. We create two variables, originalNumber and reversedNumber, to store the original number and the reversed number, respectively. We initially assign the value of number to originalNumber.
3. Next, we use a while loop to reverse the number. Inside the loop, we extract the last digit of the number using the modulo operator % and store it in the variable digit. We then update reversedNumber by multiplying it by 10 and adding the digit. Finally, we divide number by 10 to remove the last digit.
4. After the loop completes, we check if originalNumber is equal to reversedNumber. If they are equal, it means the number is a palindrome. We print a message stating that the number is a palindrome. Otherwise, if they are not equal, it means the number is not a palindrome. We print a message stating that the number is not a palindrome.

QUESTION 4) Write a java program to find the sum of n natural numbers

SOLUTION ;-

```

public class SumOfNaturalNumbers {

    public static void main(String[] args) {

        int n = 10;

        int sum = 0;

        for (int i = 1; i <= n; i++) {

            sum += i;

        }
    }
}

```

```

System.out.println("The sum of the first " + n + " natural numbers is: " + sum);

```

```
}  
  
}
```

EXPLANATION;-

1. In this program, we have set the value of n to 10.
2. We initialize the variable sum to 0, which will hold the cumulative sum of the natural numbers.
3. We use a for loop to iterate from 1 to n. In each iteration, we add the current value of i to the sum variable using the += operator. This adds the current value of i to the existing value of sum.
4. After the for loop completes, the sum variable will contain the sum of the first n natural numbers.
5. Finally, we print the result using System.out.println(), displaying the sum of the first n natural numbers.

QUESTION 5). Write a java program to Check Prime Number or not,

SOLUTION;-

```
public class CheckPrime {  
  
    public static void main(String[] args) {  
  
        int number = 13;  
  
  
        boolean isPrime = true;  
  
  
        if (number <= 1) {  
  
            isPrime = false;  
  
        } else {  
  
            for (int i = 2; i <= Math.sqrt(number); i++) {  
  
                if (number % i == 0) {  
  
                    isPrime = false;  
  
                    break;  
  
                }  
  
            }  
  
        }  
  
    }  
  
}
```

```

    }

}

if (isPrime) {

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

} else {

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

}

}

}

```

EXPLANATION;-

1. In this program, we have assigned the value 13 to the variable number.
2. We use two variables: number to store the number we want to check, and isPrime to keep track of whether the number is prime or not.
3. To determine if a number is prime, we first check if the number is less than or equal to 1. If it is, then it is not prime. In that case, we set the isPrime variable to false.
4. If the number is greater than 1, we use a for loop to iterate from $i = 2$ to $i \leq \text{Math.sqrt}(\text{number})$. Inside the loop, we check if the number is divisible evenly by i . If it is, then the number is not prime, and we set the isPrime variable to false and exit the loop using the break statement.
5. After the loop completes, we check the value of isPrime. If it is true, it means the number is prime, and we print a message stating that the number is prime. Otherwise, if it is false, it means the number is not prime, and we print a message stating that the number is not prime.