

Program 1: Find the Sum of All Even Numbers up to 50

We want to add all even numbers from 1 to 50.

Even numbers are numbers that are divisible by 2 like 2, 4, 6, ..., 50.

Explanation:

- We'll start from number 1 and go up to 50 using a **for** loop.
 - For each number, we'll check if it's even using **% 2 == 0**.
 - If it is even, we'll add it to a total.
 - At the end, we'll print the total.
-

Code:

```
total = 0    # We will keep adding even numbers into this variable
```

```
for number in range(1, 51):    # Goes from 1 to 50
```

```
    if number % 2 == 0:
```

```
        total += number
```

```
print("Sum of even numbers up to 50 is:", total)
```

Output:

Sum of even numbers up to 50 is: 650

Program 2: Print First 20 Numbers and Their Squares

Goal:

We want to print numbers from 1 to 20, and for each, print its square (number × number).

Explanation:

- We use a **for** loop from 1 to 20.
 - Inside the loop, we multiply the number by itself to find the square.
 - Then we print both.
-

```
for number in range(1, 21):           # Loop from 1 to 20
```

```
square = number * number # Find the square of the number  
print("Number:", number, "Square:", square)
```

Output (first few lines):

Number: 1 Square: 1

Number: 2 Square: 4

Number: 3 Square: 9

...

Number: 20 Square: 400

 **Program 3: Sum of First 10 Odd Numbers Using `while` Loop**

We want to add the first 10 odd numbers using a `while` loop.

Odd numbers are: 1, 3, 5, 7, ..., 19

Explanation:

- Start from 1, which is the first odd number.
 - Keep adding the number to total.
 - Move to the next odd number by doing `number += 2`.
 - Keep count of how many odd numbers we've added (`count < 10`).
 - Stop after 10 odd numbers.
-

```
count = 0          # Count of how many odd numbers we have
added
```

```
number = 1         # Start from the first odd number
```

```
total = 0          # Keep adding odd numbers here
```

```
while count < 10:   # Run loop 10 times
```

```
    total += number  # Add current odd number
```

```
    number += 2      # Move to next odd number
```

```
    count += 1       # Increase the count
```

```
print("Sum of first 10 odd numbers is:", total)
```

Output:

Sum of first 10 odd numbers is: 100

Program 4: Find Numbers Divisible by 8 and 12 (up to 100)

We want to check every number from 1 to 100, and print the numbers that are divisible by both 8 and 12.

-
- Use a **for** loop from 1 to 100.
 - For each number, check:
 - **number % 8 == 0** → divisible by 8
 - **number % 12 == 0** → divisible by 12
 - If both conditions are true, we print the number.
-

```
for number in range(1, 101):           # Loop from 1 to 100
```

```
    if number % 8 == 0 and number % 12 == 0: # Divisible by
both 8 AND 12

        print(number, "is divisible by both 8 and 12")
```

Output:

24 is divisible by both 8 and 12

48 is divisible by both 8 and 12

72 is divisible by both 8 and 12

96 is divisible by both 8 and 12

1. What is a Function?

A **function** is a reusable block of code that performs a specific task. It helps organize code and makes it easier to reuse, debug, and maintain.

Real-life Analogy:

Think of a function like a coffee machine:

- You give it inputs (water + coffee powder)
- Press the button (call the function)
- It gives you coffee (output)

```
def say_hello():
    print("Hello, welcome to Python!")
```

```
say_hello() # Calling the function
```

Creating and Calling Functions

Basic Function Syntax:

```
def function_name():  
    # code block
```

Example 1: A Greeting Function

```
def greet():  
    print("Hello, students!")
```

```
greet()
```

Example 2: Reuse Function

```
def welcome():  
    print("Welcome to Python class!")
```

```
welcome()  
welcome()
```

Functions with Parameters

♦ What are Parameters?

Parameters are variables that you pass to a function. Think of it like inputs you give to a machine.

```
def greet(name):  
    print("Hello", name)
```

```
greet("Amit")  
greet("Sara")
```

Example: Adding Numbers

```
def add(a, b):  
    print("Sum:", a + b)
```

```
add(3, 5)
add(10, 20)
```

Returning Values from Functions

What is **return**?

The **return** keyword sends a result back to the part of the code that called the function.

Example 1: Multiply Numbers

```
def multiply(a, b):
    return a * b

result = multiply(4, 5)
print("Result:", result)
```

Example 2: Check Even/Odd

```
def is_even(num):
    return num % 2 == 0

print(is_even(10)) # True
print(is_even(7)) # False
```

More Function Examples

Grade Calculator

```
def grade(marks):
    if marks >= 90:
        return "A"
    elif marks >= 75:
        return "B"
    elif marks >= 60:
        return "C"
    else:
        return "D"
```



```
print(grade(85))
```

Check Login

```
def login(username, password):  
    if username == "admin" and password == "1234":  
        return "Login Successful"  
    else:  
        return "Invalid Credentials"
```

```
print(login("admin", "1234"))
```

Practice Questions

1. Write a function to print your name
2. Create a function to add 3 numbers
3. Write a function that checks if a number is positive or negative
4. Make a function that returns the square of a number
5. Write a function `hello_user(name)` that greets a user
6. Create a function to calculate average marks of 5 subjects
7. Write a function that returns "Adult" if age ≥ 18 , else "Minor"
8. Make a function to check if a number is divisible by both 2 and 3