1. Create a dictionary of 5 students with their marks. Print the dictionary.

In [2]:

```
students_marks = {
    "Shubham": 85,
    "Ram": 92,
    "Charlie": 78,
    "Shreeman": 88,
    "Milind": 95
}
print(students_marks)
```

```
{'Shubham': 85, 'Ram': 92, 'Charlie': 78, 'Shreeman': 88, 'Milind': 95}
```

2. Access the value of a specific key from a dictionary.

In [3]:

```
students_marks = {
    "Shubham": 85,
    "Ram": 92,
    "Charlie": 78,
    "Shreeman": 88,
    "Milind": 95
}
print(students_marks["Shubham"])
```

```
85
```

3. Add a new key–value pair in the dictionary.

In [4]:

```
students_marks = {
    "Shubham": 85,
    "Ram": 92,
    "Charlie": 78,
    "Shreeman": 88,
    "Milind": 95
}
students_marks["Shubham"] = 80

print(students_marks)
```

```
{'Shubham': 80, 'Ram': 92, 'Charlie': 78, 'Shreeman': 88, 'Milind': 95}
```

4. Update the value of an existing key in the dictionary.

In [5]:

```
students_marks = {
    "Shubham": 85,
    "Ram": 92,
    "Charlie": 78,
    "Shreeman": 88,
    "Milind": 95
}
students_marks["Shreeman"] = 82
print(students_marks)
```

```
{'Shubham': 85, 'Ram': 92, 'Charlie': 78, 'Shreeman': 82, 'Milind': 95}
```

5. Remove a key–value pair from the dictionary using pop().

In [6]:

```
students_marks = {
    "Shubham": 85,
    "Ram": 92,
    "Charlie": 78,
    "Shreeman": 88,
    "Milind": 95
}
students_marks.pop("Milind")
print(students_marks)
```

```
{'Shubham': 85, 'Ram': 92, 'Charlie': 78, 'Shreeman': 88}
```

6. Get all keys of a dictionary using .keys() function.

In [7]:

```
students_marks = {
    "Shubham": 85,
    "Ram": 92,
    "Charlie": 78,
    "Shreeman": 88,
    "Milind": 95
}
keys = students_marks.keys()
print(keys)
```

```
dict_keys(['Shubham', 'Ram', 'Charlie', 'Shreeman', 'Milind'])
```

7. Get all values of a dictionary using .values() function.

In [8]:

```
students_marks = {
    "Shubham": 85,
    "Ram": 92,
    "Charlie": 78,
    "Shreeman": 88,
    "Milind": 95
}
values = students_marks.values()
print(values)
```

```
dict_values([85, 92, 78, 88, 95])
```

8. Get all key–value pairs from a dictionary using .items().

In [9]:

```
students_marks = {
    "Shubham": 85,
    "Ram": 92,
    "Charlie": 78,
    "Shreeman": 88,
    "Milind": 95
}
items = students_marks.items()
print(items)
```

```
dict_items([('Shubham', 85), ('Ram', 92), ('Charlie', 78), ('Shreeman', 88), ('Milin
d', 95)])
```

9. Check if a given key exists in the dictionary.

In [10]:

```
students_marks = {
    "Shubham": 85,
    "Ram": 92,
    "Charlie": 78,
    "Shreeman": 88,
    "Milind": 95
}
if "Ram" in students_marks:
    print("Key exists")
else:
    print("Key does not exist")
```

```
Key exists
```

10. Create a dictionary and print only keys using a loop.

In [11]:
```
students_marks = {
    "Shubham": 85,
    "Ram": 92,
    "Charlie": 78,
    "Shreeman": 88,
    "Milind": 95
}
for key in students_marks:
    print(key)
```

```
Shubham
Ram
Charlie
Shreeman
Milind
```

11. Create a dictionary and print only values using a loop.

In [12]:
```
students_marks = {
    "Shubham": 85,
    "Ram": 92,
    "Charlie": 78,
    "Shreeman": 88,
    "Milind": 95
}

for value in students_marks.values():
    print(value)
```

```
85
92
78
88
95
```

12. Merge two dictionaries.

In [13]:
```
students_marks1 = {
    "Shubham": 85,
    "Sumit": 92
}
students_marks2 = {
    "Charlie": 78,
    "Shreeman": 88
}
```

```
merged_dict = {students_marks1, students_marks2}

print(merged_dict)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[13], line 10
      1 students_marks1 = {
      2     "Alice": 85,
      3     "Bob": 92
      4 }
      5 students_marks2 = {
      6     "Charlie": 78,
      7     "David": 88
      8 }
---> 10 merged_dict = {students_marks1, students_marks2}
     12 print(merged_dict)

TypeError: unhashable type: 'dict'
```

13. Write a program to find the key with the maximum value in a dictionary.

In [14]:

```
students_marks = {
    "Shubham": 85,
    "Ram": 92,
    "Charlie": 78,
    "Shreeman": 88,
    "Milind": 95
}
max_key = max(students_marks, key=students_marks.get)
print("Key with maximum value:", max_key)
```

```
Key with maximum value: Milind
```

14. Create a nested dictionary and access the inner dictionary value.

In [15]:

```
students_info = {
    "Ram": {"marks": 85, "age": 20},
    "Sham": {"marks": 92, "age": 21}
}

bob_marks = students_info["Sham"]["marks"]
print("Sham's Marks:", bob_marks)
```

```
Sham's Marks: 92
```

15. Create two lists: one with names and one with marks. Use zip() to combine them into a dictionary.

In [17]:

```
names = ["Shubham", "vikas", "Charlie", "Sham", "Ram"]
marks = [85, 92, 78, 88, 95]
students_marks = dict(zip(names, marks))

print(students_marks)
```

```
{'Shubham': 85, 'vikas': 92, 'Charlie': 78, 'Sham': 88, 'Ram': 95}
```

16. Create a dictionary using fromkeys() with a list of keys and the same default value.

In [18]:

```
keys = ["Shubham", "vikas", "Charlie", "Sham", "Ram"]
students_marks = dict.fromkeys(keys, 0)

print(students_marks)
```

```
{'Shubham': 0, 'vikas': 0, 'Charlie': 0, 'Sham': 0, 'Ram': 0}
```

In [ ]:

1. Write a function to print "Hello, World!".

In [1]:

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

greet()
```

```
Hello, World!
```

2. Write a function that takes your name as input and prints a greeting message.

In [2]:

```
def greet_name(name):
    print("Hello,", name)
greet_name("Shubham")
```

```
Hello, Shubham
```

3. Write a function to find the sum of two numbers (take input from user).

In [3]:

```
def add_numbers():
    num1 = float(input("Enter first number: "))
    num2 = float(input("Enter second number: "))
    total = num1 + num2
    print("Sum:", total)

add_numbers()
```

```
Sum: 11.0
```

4. Write a function to return the square of a number.

In [4]:

```
def square(num):
    return num * num

result = square(5)
print("Square:", result)
```

```
Square: 25
```

5. Write a function to check if a number is even or odd.

In [8]:

```
def check_even_odd(num):
    if num % 2 == 0:
        print("Even")
    else:
        print("Odd")

check_even_odd(7)
```

```
Odd
```

6. Write a function to find the factorial of a given number.

In [12]:

```
def factorial(n):
    fact = 1
    for i in range(1, n + 1):
        fact *= i
    return fact
```

```
num = 5
print("Factorial of", num, "is", factorial(num))
```

```
Factorial of 5 is 120
```

7. Write a function that takes a list and returns the sum of all its elements.

In [13]:

```
def sum_list(numbers):
    return sum(numbers)
my_list = [1, 2, 3, 4, 5]
print("Sum of list elements:", sum_list(my_list))
```

```
Sum of list elements: 15
```

8. Write a function that takes a list and returns a new list with only even numbers.

In [14]:

```
def get_even_numbers(numbers):
    even_list = []
    for num in numbers:
        if num % 2 == 0:
            even_list.append(num)
    return even_list

my_list = [1, 2, 3, 4, 5, 6]
print("Even numbers:", get_even_numbers(my_list))
```

```
Even numbers: [2, 4, 6]
```

9. Write a function that takes two strings and returns the longer one.

In [15]:

```
def longer_string(str1, str2):
    if len(str1) > len(str2):
        return str1
    else:
        return str2
print("Longer string:", longer_string("Hello", "World!"))
```

```
Longer string: World!
```

10. Write a function that converts Celsius temperature to Fahrenheit.

In [2]:

```
def celsius_to_fahrenheit(celsius):
    return (celsius * 4/5) + 30
```

```
temp_c = 25
print("Temperature in Fahrenheit:", celsius_to_fahrenheit(temp_c))
```

```
Temperature in Fahrenheit: 50.0
```

In [ ]: