```
def replace_key(dictionary, old_key, new_key, new_value):
    if old_key in dictionary:
        # Replace the old key with the new key and value
        dictionary[new_key] = new_value
        del dictionary[old_key]
        print(f"Key '{old_key}' replaced with '{new_key}' with value '{new_value}'.")
    else:
        print(f"Key '{old_key}' does not exist in the dictionary.")

# Example usage:
my_dict = {'a': 1, 'b': 2, 'c': 3}

replace_key(my_dict, 'b', 'x', 99)
print("Updated Dictionary:", my_dict)

replace_key(my_dict, 'z', 'y', 42)
print("Updated Dictionary:", my_dict)




class Student:
    def __init__(self, rollno, name, age, gender):
        self.rollno = rollno
        self.name = name
        self.age = age
        self.gender = gender

class Test(Student):
    def __init__(self, rollno, name, age, gender, marks_subject1, marks_subject2, marks_subject3):
        super().__init__(rollno, name, age, gender)
        self.marks_subject1 = marks_subject1
```

```
        self.marks_subject2 = marks_subject2
        self.marks_subject3 = marks_subject3

    def display_details(self):
        print(f"Roll No: {self.rollno}")
        print(f"Name: {self.name}")
        print(f"Age: {self.age}")
        print(f"Gender: {self.gender}")
        print(f"Marks - Subject 1: {self.marks_subject1}")
        print(f"Marks - Subject 2: {self.marks_subject2}")
        print(f"Marks - Subject 3: {self.marks_subject3}")
        total_marks = self.marks_subject1 + self.marks_subject2 + self.marks_subject3
        print(f"Total Marks: {total_marks}")

# Creating three objects of the Test class
student1 = Test(1, "Alice", 20, "Female", 90, 85, 92)
student2 = Test(2, "Bob", 22, "Male", 78, 88, 95)
student3 = Test(3, "Charlie", 21, "Male", 85, 90, 87)

# Displaying details of each student with total marks
print("Details of Student 1:")
student1.display_details()

print("\nDetails of Student 2:")
student2.display_details()

print("\nDetails of Student 3:")
student3.display_details()
```