CODES GENARATED BY GITHUB COPOILET:-

TASK1:-

Code:-

```
class BankAccount:
    def __init__(self, account_holder, balance=0.0):
        self.account_holder = account_holder
        self.balance = balance

    def deposit(self, amount):
        if amount > 0:
            self.balance += amount
            print(f"Deposited {amount}. New balance: {self.balance}")
        else:
            print("Deposit amount must be positive.")

    def withdraw(self, amount):
        if amount > 0:
            if self.balance >= amount:
                self.balance -= amount
                print(f"Withdrew {amount}. New balance: {self.balance}")
            else:
                print("Insufficient funds.")
        else:
            print("Withdrawal amount must be positive.")

    def display_balance(self):
        print(f"Account holder: {self.account_holder}, Balance: {self.balance}")

if __name__ == "__main__":
    account = BankAccount("Alice", 100)
    account.display_balance()
    account.deposit(50)
    account.withdraw(30)
    account.display_balance()
```

Output:-

```
Account holder: Alice, Balance: 100
Deposited 50. New balance: 150
Withdrew 30. New balance: 120
Account holder: Alice, Balance: 120
```

TASK 2:-

CODE:-

```
numbers = [1, 2, 3, 4, 5, 6]
total = 0
for num in numbers:
    if num % 2 == 0:
        total += num

print("Sum of even numbers:", total)
```

Output:-

```
>>> %Run -c $EDITOR_CONTENT
  Sum of even numbers: 12
```

TASK 3:-

CODE:-

```
def age_group(age):
    if age < 13:
        return "Child"
    elif age < 20:
        return "Teenager"
    elif age < 65:
        return "Adult"
    else:
        return "Senior"
s1=age_group(5)
print(s1)
```

OUTPUT:-

```
Child
```

TASK 4:-

CODE:-

```
number = 1234
reversed_num = 0

while number > 0:
    digit = number % 10
    reversed_num = reversed_num * 10 + digit
    number //= 10

print(reversed_num)
```

OUTPUT:-

```
4321
```

TASK 5:-

Code:-

```python
# Base class Employee
class Employee:
    def __init__(self, name, salary):
        self.name = name
        self.salary = salary

    def display(self):
        print(f"Name: {self.name}, Salary: {self.salary}")

# Derived class Manager
class Manager(Employee):
    def __init__(self, name, salary, department):
        super().__init__(name, salary)
        self.department = department

    def display(self):
        print(f"Name: {self.name}, Salary: {self.salary}, Dept: {self.department}")

# Example usage
mgr = Manager("John", 50000, "IT")
mgr.display()  # Output: Name: John, Salary: 50000, Dept: IT
```

OUTPUT:-

```
>>> %Run -c $EDITOR_CONTENT
  Name: John, Salary: 50000, Dept: IT
```