|  |  |
| --- | --- |
| EX:NO:4c | Implementing real-time/technical applications using Lists, Tuples. Materials required for construction of a building |
| Date: |

**Aim:**

To Implement real-time/technical applications using Lists, Tuples. Items Materials required for construction of a building

**Algorithm:**

Step1: Start the program

Step2: Assign list of building materials

Step3: Perform all the list operation

Step4: print the result

Step5: End the program

**Program:**

class BuildingConstruction:

def \_\_init\_\_(self):

self.materials = []

def add\_material(self, material\_name, quantity, unit):

material = (material\_name, quantity, unit)

self.materials.append(material)

print(f"Added {quantity} {unit} of {material\_name}")

def display\_materials(self):

if not self.materials:

print("No materials added for construction.")

else:

print("Materials Required for Construction:")

for material in self.materials:

material\_name, quantity, unit = material

print(f"{quantity} {unit} of {material\_name}")

def main():

construction = BuildingConstruction()

while True:

print("\nBuilding Construction Material Management System")

print("1. Add Material")

print("2. Display Materials")

print("3. Exit")

choice = input("Enter your choice: ")

if choice == '1':

material\_name = input("Enter material name: ")

quantity = float(input("Enter quantity: "))

unit = input("Enter unit (e.g., bags, tons, etc.): ")

construction.add\_material(material\_name, quantity, unit)

elif choice == '2':

construction.display\_materials()

elif choice == '3':

print("Goodbye!")

break

else:

print("Invalid choice. Please try again.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Result:**