```
# Define a class to represent a parcel
class Parcel:
    def __init__(self, weight, profit):
        self.weight = weight
        self.profit = profit
        self.profit_per_weight = profit / weight  # Calculate profit-to-weight ratio

# Function to maximize profit using the Fractional Knapsack strategy
def fractional_knapsack(parcels, capacity):
    # Sort parcels by profit-to-weight ratio in descending order
    parcels.sort(key=lambda x: x.profit_per_weight, reverse=True)

    total_profit = 0.0  # Total profit accumulated
    for parcel in parcels:
        if capacity == 0:  # If the truck is full, stop
            break
        if parcel.weight <= capacity:
            # Take the entire parcel
            total_profit += parcel.profit
            capacity -= parcel.weight
        else:
            # Take a fraction of the parcel
            total_profit += parcel.profit_per_weight * capacity
            capacity = 0  # Truck is now full

    return total_profit

# Main function to test the program
if __name__ == "__main__":
```

```
# Input: List of parcels with weights and profits
parcels = [
    Parcel(10, 60),  # Parcel 1: weight=10, profit=60
    Parcel(20, 100), # Parcel 2: weight=20, profit=100
    Parcel(30, 120)  # Parcel 3: weight=30, profit=120
]
truck_capacity = 50  # Maximum weight capacity of the truck

# Calculate the maximum profit
max_profit = fractional_knapsack(parcels, truck_capacity)
print(f"Maximum profit that can be achieved: {max_profit:.2f}")
```