```
// Program 1A
// CIS 200-01/76
// Fall 2017
// Due: 9/25/2017
// By: C5503

// File: TestParcels.cs
// This is a simple, console application designed to exercise the Parcel hierarchy.
// It creates several different Parcels and prints them.

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Prog1
{
    class TestParcels
    {
        // Precondition:  None
        // Postcondition: Parcels have been created and displayed
        static void Main(string[] args)
        {
            // Test Data - Magic Numbers OK
            Address a1 = new Address("  John Smith  ", "   123 Any St.   ", "  Apt. 45 ",
                "  Louisville   ", "  KY   ", 40202); // Test Address 1
            Address a2 = new Address("Jane Doe", "987 Main St.",
                "Beverly Hills", "CA", 90210); // Test Address 2
            Address a3 = new Address("James Kirk", "654 Roddenberry Way", "Suite 321",
                "El Paso", "TX", 79901); // Test Address 3
            Address a4 = new Address("John Crichton", "678 Pau Place", "Apt. 7",
                "Portland", "ME", 04101); // Test Address 4
            Address a5 = new Address("Travis Scott", "3845 Straight Up Avenue", "Apt. 8"
                , "Houston", "TX", 77001); // Test Address 5
            Address a6 = new Address("Kendrick Lamar", "2201 Compton Road",
                "Compton", "CA", 90059); // Test Address 6
            Address a7 = new Address("Jaden Smith", "584 Smith Way",
                "Los Angeles", "CA", 90001); // Test Address 7
            Address a8 = new Address("Paul DeNino", "4304 Poseidon Road",
                "Miami", "FL", 33018); // Test Address 8

            Letter letter1 = new Letter(a1, a2, 3.95M);  // Letter test objects
            Letter letter2 = new Letter(a5, a6, 1.40M);

            GroundPackage gp1 = new GroundPackage(a3, a4, 14, 10, 5, 12.5);   // Ground
test objects
            GroundPackage gp2 = new GroundPackage(a7, a8, 14, 9, 4, 10);

            NextDayAirPackage ndap1 = new NextDayAirPackage(a1, a3, 25, 15, 15,   // Next
Day test objects
                85, 7.50M);
            NextDayAirPackage ndap2 = new NextDayAirPackage(a3, a6, 20, 12, 10,
                70, 8.00M);

            TwoDayAirPackage tdap1 = new TwoDayAirPackage(a4, a7, 46.5, 39.5, 28.0, //
Two Day test objects
                80.5, TwoDayAirPackage.Delivery.Saver);
            TwoDayAirPackage tdap2 = new TwoDayAirPackage(a6, a8, 47.8, 29.3, 32.0,
```

```
            75.0, TwoDayAirPackage.Delivery.Early);

        List<Parcel> parcels;      // List of test parcels

        parcels = new List<Parcel> // Adding Address objects to list parcels
        {
            letter1,
            letter2,
            gp1,
            gp2,
            ndap1,
            ndap2,
            tdap1,
            tdap2
        };
        // Displays the original list of parcels in console
        Console.WriteLine("Original List:");
        Console.WriteLine("====================");
        foreach (Parcel p in parcels)
        {
            Console.WriteLine(p);
            Console.WriteLine("====================");
        }
        Pause();

        // Use LINQ to select all Parcels and order by Destination Zip Code in
descending order
        var parcelsOrderedByDestZip =
            from p in parcels
            orderby p.DestinationAddress.Zip descending
            select p;

        // Displays the results for the above in console
        Console.WriteLine("Parcels by Destination Zip Code Descending:");
        Console.WriteLine("====================");
        foreach (Parcel p in parcelsOrderedByDestZip)
        {
            Console.WriteLine(p);
            Console.WriteLine("====================");
        }
        Pause();

        // Use LINQ to select all Parcels and order by Cost in ascending order
        var parcelsOrderedByCost =
            from p in parcels
            orderby p.CalcCost() ascending
            select p;

        // Displays the results for the above in console
        Console.WriteLine("Parcels by Cost Ascending:");
        Console.WriteLine("====================");
        foreach (Parcel p in parcelsOrderedByCost)
        {
            Console.WriteLine(p);
            Console.WriteLine("====================");
        }
        Pause();
```

```
            // Use LINQ to select all Parcels and order by Parcel type in ascending order
then Cost in descending order
            var parcelsOrderedByTypeAndCost =
                from p in parcels
                orderby p.GetType().ToString() ascending, p.CalcCost() descending
                select p;

            // Displays the results for the above in console
            Console.WriteLine("Parcels by Parcel Type Ascending and then Cost
Descending:");
            Console.WriteLine("====================");
            foreach (Parcel p in parcelsOrderedByTypeAndCost)
            {
                Console.WriteLine(p);
                Console.WriteLine("====================");
            }
            Pause();

            // Use LINQ to select all AirPackage objects that are considered Heavy and
order by Weight in descending order
            var airPackageIsHeavyAndOrderedByWeight =
                from p in parcels
                where p is AirPackage && ((AirPackage)p).IsHeavy() == true
                orderby ((AirPackage)p).Weight descending
                select p;

            // Displays the results for the above in console
            Console.WriteLine("AirPackages that are Heavy and Ordered by Weight
Descending:");
            Console.WriteLine("====================");
            foreach (Parcel p in airPackageIsHeavyAndOrderedByWeight)
            {
                Console.WriteLine(p);
                Console.WriteLine("====================");
            }
            Pause();
        }

        // Precondition:  None
        // Postcondition: Pauses program execution until user presses Enter and
        //                then clears the screen
        public static void Pause()
        {
            Console.WriteLine("Press Enter to Continue...");
            Console.ReadLine();

            Console.Clear(); // Clear screen
        }
    }
}
```