Scanner in = new Scanner(System.in);
System.out.print("Enter Quantity: ");
int quantity = in.nextInt();
System.out.print("Enter Price: ");
double price = in.nextDouble();
System.out.print("Enter City: ");
String city = in.nextLine(); // until end of file
System.out.print("Enter State Code: ");
String code = in.next(); // until white space

The file name is InputTester.

import java.util.Scanner;
public class InputTester{
public static void main(String[] args){
Scanner in = new Scanner(System.in);
CashRegister register = new CashRegister();
System.out.print("Enter Price: ");
double price = in.nextDouble();
System.out.print("Enter Quarters: ");
int quarters = in.nextInt();
System.out.print("Enter Dimes: ");
int dimes = in.nextInt();
System.out.print("Enter Nickels: ");
int nickels = in.nextInt();
System.out.print("Enter Pennies: ");
int pennies = in.nextInt();
register.enterPayment(price, quarters, dimes, nickels, pennies);
System.out.print("Your Change is: ");
System.out.println(register.giveChange());

Inheritance

public class Attraction {
    // Define instance variable:
    public int minutes;
    // Define zero-parameter constructor:
    public Attraction () {
        System.out.println("Calling Attraction constructor");
        minutes = 75;
    }
    public Attraction (int m) {minutes = m;}
}

public class Concert extends Attraction {
public void onSale() {
} // public void onSale()

public class Cafe extends Attraction {
public void onSale() {
} // public void onSale()

The file name is Inheritance.java.

Reading Input/Java 5.0 (cont)
Inheritance Continued

public class Movie extends Attraction {

    public int script, acting, direction;

    public Movie () {
        System.out.println("Calling Movie constructor");
        script = 5; acting = 5; direction = 5;
    }

    public Movie (int s, int a, int d) {
        script = s; acting = a; direction = d;
    }

    public int rating () {
        return script + acting + direction;
    }

}

Inheritance Continued

public class Symphony extends Attraction {

    public int music, playing, conducting;

    public Symphony () {
        System.out.println("Calling Symphony constructor");
        music = 5; playing = 5; conducting = 5;
    }

    public Symphony (int m, int p, int c) {
        music = m; playing = p; conducting = c;
    }

    public int rating () {
        return music + playing + conducting;
    }

}
### Arrays

```java
int durations[];
new int[4];
durations[] = new int[4];
durations[] = {65, 87, 72, 75};
durations[3] = 65;
durations[] = {65, 87, 72, 75};
durations[] = new int[4];
durations[] = [4];
durations[] = [];

Arrays
```
```java
public class Vehicle {
    public void print() {
        System.out.println("A Vehicle");
    }
}

public class MotorVehicle extends Vehicle {
    String regNum;
    public MotorVehicle(String no) {
        regNum = no;
    }
    public void print() {
        System.out.println("A Motor Vehicle with reg no: "+regNum);
    }
}

public class PrivateCar extends MotorVehicle {
    int numSeats;
    public PrivateCar(String no, int n) {
        super(no);
        numSeats = n;
    }
    public void print() {
        super.print();
        System.out.println("Private car with "+numSeats + " seats");
    }
}
```

Arrays Example (cont)

```java
ArrayLists Example
```
Using Arrays (cont)

Vehicle[] veh = new Vehicle;
veh[0] = new PrivateCar("ABC123", 5);
veh[1] = new Truck("XYZ999", 10000);
veh[2] = new PrivateCar("PPP000", 6);
veh[3] = new Bike(10);

for (int i = 0; i < veh.length; i++)
    if (veh[i] != null)
        veh[i].print();

ArrayLists Example (cont)

public class Truck extends MotorVehicle {
    int maxL;
    public Truck(String no, int load) {
        super(no);
        maxL = load;
    }
    public void print() {
        super.print();
        System.out.println("A Truck with with: "+maxL + " kg maximum load");
    }
}

public class Bike extends Vehicle {
    String numGears;
    public Bike(int g) {
        numGears = g;
    }
    public void print() {
        System.out.println("A bike with:"+numGears + " A bike with : ");
    }
}

public class Vehicle {
    String type;
    public void print() {
        System.out.println("Vehicle with:");
    }
}

public class MotorVehicle extends Vehicle {
    public void print() {
        super.print();
        System.out.println("Motor vehicle");
    }
}

public class Truck extends MotorVehicle {
    int maxL;
    public Truck(String no, int load) {
        super(no);
        maxL = load;
    }
    public void print() {
        super.print();
        System.out.println("A Truck with with:");
    }
}

public class Bike extends Vehicle {
    String numGears;
    public Bike(int g) {
        numGears = g;
    }
    public void print() {
        System.out.println("A bike with:");
    }
}

public class MotorVehicle extends Vehicle {
    public void print() {
        super.print();
        System.out.println("Motor vehicle");
    }
}

public class Truck extends MotorVehicle {
    int maxL;
    public Truck(String no, int load) {
        super(no);
        maxL = load;
    }
    public void print() {
        super.print();
        System.out.println("A Truck with with:");
    }
}

public class Bike extends Vehicle {
    String numGears;
    public Bike(int g) {
        numGears = g;
    }
    public void print() {
        System.out.println("A bike with:");
    }
}

public class MotorVehicle extends Vehicle {
    public void print() {
        super.print();
        System.out.println("Motor vehicle");
    }
}

Using Arrays (cont)

Vehicle[] veh = new Vehicle;
veh[0] = new PrivateCar("ABC123", 5);
veh[1] = new Truck("XYZ999", 10000);
veh[2] = new PrivateCar("PPP000", 6);
veh[3] = new Bike(10);

for (int i = 0; i < veh.length; i++)
    if (veh[i] != null)
        veh[i].print();

ArrayLists Example (cont)

public class Truck extends MotorVehicle {
    int maxL;
    public Truck(String no, int load) {
        super(no);
        maxL = load;
    }
    public void print() {
        super.print();
        System.out.println("A Truck with with: "+maxL + " kg maximum load");
    }
}

public class Bike extends Vehicle {
    String numGears;
    public Bike(int g) {
        numGears = g;
    }
    public void print() {
        System.out.println("A bike with:"+numGears + " A bike with : ");
    }
}

public class Vehicle {
    String type;
    public void print() {
        System.out.println("Vehicle with:");
    }
}

public class MotorVehicle extends Vehicle {
    public void print() {
        super.print();
        System.out.println("Motor vehicle");
    }
}

public class Truck extends MotorVehicle {
    int maxL;
    public Truck(String no, int load) {
        super(no);
        maxL = load;
    }
    public void print() {
        super.print();
        System.out.println("A Truck with with:");
    }
}

public class Bike extends Vehicle {
    String numGears;
    public Bike(int g) {
        numGears = g;
    }
    public void print() {
        System.out.println("A bike with:");
    }
}

public class MotorVehicle extends Vehicle {
    public void print() {
        super.print();
        System.out.println("Motor vehicle");
    }
}

Using Arrays (cont)

Vehicle[] veh = new Vehicle;
veh[0] = new PrivateCar("ABC123", 5);
veh[1] = new Truck("XYZ999", 10000);
veh[2] = new PrivateCar("PPP000", 6);
veh[3] = new Bike(10);

for (int i = 0; i < veh.length; i++)
    if (veh[i] != null)
        veh[i].print();
ArrayList u = new ArrayList();

u.add(new PrivateCar("ABC123", 5));
u.add(new Truck("XYZ999", 10000));
u.add(new PrivateCar("PPP000", 6));
u.add(new Bike(10));

for (int i = 0; i < u.size; i++)
    if (u.elementAt(i) != null)
        u.elementAt(i).print();

System.out.println();