Chapter 4

Conditionals and Control Flow

4.1 If-Then

The if-then statement is the most simple control flow we can write. It tests an expression for truth and executes some code based on it.

Example:

```
public class Order {

  public static void main(String[] args) {

    double itemCost = 30.99;

    // Write an if-then statement:
    if (itemCost > 24.00)
       System.out.println("High value item!");

  }

}
```

4.2 If-Then-Else

In our next example, we will add more attributes and methods to our Order class. We need to calculate the shipping costs for our orders. The attribute shipping is what we will use to calculate the cost. Use a chained if-then-else to check for different values within the calculateShipping() method:

- When the shipping instance field equals "Regular", the method should return 0.
- When the shipping instance field equals "Express", the method should return 1.75.
- Else the method should return .50.

```
public class Order {
  boolean isFilled;
  double billAmount;
  String shipping;


  public Order(boolean filled, double cost, String shippingMethod) {
           if (cost > 24.00) {
           System.out.println("High value item!");
           }
    isFilled = filled;
```

```
    billAmount = cost;
    shipping = shippingMethod;
  }

  public void ship() {
    if (isFilled) {
      System.out.println("Shipping");
      System.out.println("Shipping cost: " + calculateShipping());
    } else {
      System.out.println("Order not ready");
    }
  }

  public double calculateShipping() {
          // declare conditional statement here
    if (shipping == "Regular")
      return 0;
    else if (shipping == "Express")
      return 1.75;
    else
      return 0.50;
     }

  public static void main(String[] args) {
    Order book = new Order(true, 9.99, "Express");
    Order chemistrySet = new Order(false, 72.50, "Regular");
    book.ship();
    chemistrySet.ship();
  }
}
```

4.3 Switch

An alternative to chaining if-then-else conditions together is to use the switch statement. This conditional will check a given value against any number of conditions and run the code block where there is a match.

We'll rewrite the calculateShipping() method so it uses a switch statement instead.

Example:

```
public class Order {
  boolean isFilled;
  double billAmount;
  String shipping;

  public Order(boolean filled, double cost, String shippingMethod) {
            if (cost > 24.00) {
      System.out.println("High value item!");
    }
    isFilled = filled;
    billAmount = cost;
    shipping = shippingMethod;
  }

  public void ship() {
    if (isFilled) {
      System.out.println("Shipping");
      System.out.println("Shipping cost: " + calculateShipping());
    } else {
      System.out.println("Order not ready");
    }
  }

  public double calculateShipping() {
    double shippingCost;
           // declare switch statement here
    switch (shipping) {

      case "Regular":
        shippingCost = 0;
        break;
      case "Express":
        shippingCost = 1.75;
        break;
      default:
        shippingCost = .50;
    }

    return shippingCost;
      }

  public static void main(String[] args) {
    // do not alter the main method!
    Order book = new Order(true, 9.99, "Express");
    Order chemistrySet = new Order(false, 72.50, "Regular");

    book.ship();
    chemistrySet.ship();
  }
}
```

4.4 Conditional Operators

Java includes operators that only use boolean values. These conditional operators help simplify complex boolean relationships by reducing multiple boolean values to a single value:

- For example, what if we want to run a code block only if multiple conditions are true. We could use the AND operator: &&.
- If we want to run a code block if at least one of two conditions are true. We could use the OR operator: ||.
- Finally, we can produce the opposite value, where true becomes false and false becomes true, with the NOT operator: !.

Example:

```
public class Reservation {
  int guestCount;
  int restaurantCapacity;
  boolean isRestaurantOpen;
  boolean isConfirmed;

  public Reservation(int count, int capacity, boolean open) {
    if (count < 1 || count > 8) {
      System.out.println("Invalid reservation!");
    }
    guestCount = count;
          restaurantCapacity = capacity;
          isRestaurantOpen = open;
  }

  public void confirmReservation() {
    if (restaurantCapacity >= guestCount && isRestaurantOpen) {
      System.out.println("Reservation confirmed");
      isConfirmed = true;
    } else {
      System.out.println("Reservation denied");
               isConfirmed = false;
    }
  }

  public void informUser() {
    if (!isConfirmed) {
      System.out.println("Unable to confirm reservation, please
contact restaurant.");
    } else {
      System.out.println("Please enjoy your meal!");
    }
  }

  public static void main(String[] args) {
    Reservation partyOfThree = new Reservation(3, 12, true);
    Reservation partyOfFour = new Reservation(4, 3, true);
    partyOfThree.confirmReservation();
```

```
    partyOfThree.informUser();
    partyOfFour.confirmReservation();
    partyOfFour.informUser();
  }
}
```