Conditionals

```java
import java.util.Scanner;

public class MaxCalculator {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("a=");
        int a = sc.nextInt();
        System.out.print("b=");
        int b = sc.nextInt();
        System.out.print("c=");
        int c = sc.nextInt();

        // find max of a,b,c
        int max = 0;
        if (a > b) {
            max = a;
            if (c > max)
                max = c;
        } else {
            max = b;
            if (c > max)
                max = c;
        }

        System.out.println("max = " + max);
    }
}
```

The switch statement

1. class OlympicGamesStats {
   String retrieveCity(int year) {
       String city;
switch(year) {
    case 1992:
        city = "Barcelona";
        break;
    case 1996:
        city = "Atlanta";
        break;
    case 2000:
        city = "Sydney";
        break;
    case 2004:
        city = "Athens";
        break;
    default:
        city = "unknown";
} 
return city;

2. import java.util.Scanner;

public class OlympicGamesStatsTest {
    public static void main(String[] args) {
        OlympicGamesStats stats = new OlympicGamesStats();

        Scanner sc = new Scanner(System.in);
        System.out.print("Enter an year:");
        int year = sc.nextInt();

        String city = stats.retrieveCity(year);
        System.out.println("City which organised Olympic Games in " + year + ": " + city);
    }
}

For Loops

1. class Coin {
    public String flip() {
        String result;

        if (Math.random() <= 0.5)
            result = "head";
        else
            result = "tail"; 
}
2. public class CoinFlipper {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of flips: ");
        int times = sc.nextInt();

        Coin c = new Coin();
        int number_of_heads = 0;
        int number_of_tails = 0;

        for (int i=1; i <= times; i++) {
            String result = c.flip();
            if (result.equals("head"))
                ++number_of_heads;
            else
                ++number_of_tails;
        }

        System.out.println("Number of heads: "+number_of_heads);
        System.out.println("Number of tails: "+number_of_tails);
    }
}

While loops

1. class SellStocks {
    private double price;
    private int number_of_stocks;
    private double commission_rate;

    SellStocks(double price, int number_of_stocks, double commission_rate) {
        this.price = price;
        this.number_of_stocks = number_of_stocks;
        this.commission_rate = commission_rate;
    }

    public double calculate() {
        double total = price*number_of_stocks;
        total = total - total*commission_rate/100;

        return total;
    }
}

2. public class SellStocksTest {
public static void main(String[] args) {
    double price = -1;
    Scanner sc = new Scanner(System.in);

    while (price != 0) {
        System.out.print("Enter price: ");
        price = sc.nextDouble();
        if (price == 0)
            break;
        System.out.print("Enter number of stocks: ");
        int number = sc.nextInt();
        System.out.print("Enter commission rate: ");
        double commission_rate = sc.nextDouble();

        SellStocks st = new SellStocks(price, number, commission_rate);
        double amount = st.calculate();

        // display the amount obtained from selling stocks
        System.out.println("Amount from stock selling: "+ amount);
    }
}

Converting between different types of loops

1. public class Square {
   public static void main(String[] args) {
       int n = 10;
       int i = 1;
       while (i <= n) {
           System.out.println(i + " squared equals "+ i * i);
           ++i;
       }
   }
}

2. public class Square {
   public static void main(String[] args) {
       int n = 10;
       int i = 1;
       do {
           System.out.println(i + " squared equals "+ i * i);
           ++i;
       } while (i <= n);
   }
}
Nested loops

```java
public class SodaCanDrawer {
    public static void main(String[] args) {
        for (int i=1; i <= 4; i++) {
            for (int j=1; j <= 6; j++)
                System.out.print("O");
            System.out.println(); // insert a new line after each row
        }
    }
}
```

Break and Continue

The output of the program is:

```
j = 1
j = 3
j = 4
j = 1
j = 3
j = 4
j = 1
j = 3
j = 4
```

The null keyword

During execution an exception occurs:

```
Exception in thread "main" java.lang.NullPointerException
    at NullObjectsTest.main(NullObjectsTest.java:4)
```

A reference variable must point to an object before attempting to access fields or call methods on the object. Since `s` contains the value `null`, method `toUpperCase()` cannot be called on `null` and an exception will be generated during running the program.