Chapter 18 - Part 2

Recursion
Code Examples

Example: Computing Factorial

```
//Return the factorial for a specified number
public class Factorial
{
   public static void main(String[] args)
   {
      //print factorial of 5
      long fact_5 = factorial(5);
      System.out.println(fact_5);
   }

   public static long factorial(int n)
   {
      if (n == 0) //Base case
         return 1;
      else
         return n * factorial(n - 1); //Recursive call
   }
}
```

Example: Computing Sum

```
//Return the sum of all number form 1 to for a specified
//number, say n.
public class RecursiveSum
{
   public static void main(String[] args)
   {
      //print sum of 5
      int sum_5 = sum(5);
      System.out.println(sum_5);
   }

   public static int sum (int n)
   {
      if (n == 1) //Base case
         return 1;
      else
         return n + sum(n - 1); //Recursive call
   }
}
```

Example: Computing Fibonacci Sequence

```
public class RecursiveFib //Return the fibonacci sequence
{
   public static void main(String[] args)
   {  //print fibonancci sequence value for index 10
      long fib_10 = fib(10);
      System.out.println(fib_10);
      System.out.println();
      //print fibonancci sequence up to index 10
      for (int i=0; i<=10; i++)
         System.out.print(fib(i) + "  ");
   }

   public static long fib(long index)
   {  if (index == 0) //Base case
         return 0;
      else if (index == 1)
         return 1;
      else
         return fib(index-1) + fib(index-2); //Recursive call
   }
}
```

Example: Print a Message *n* Times

```
//Print a message recursively n times . Two versions
public class PrintMessage
{
   public static void main(String[] args)
   {  //print "Hello World" 5 times
      String message = "Hello World!";
      nPrintln_v1(message,5); }

   //Version 1
   public static void nPrintln_v1(String message, int times)
   {  if (times >= 1){
         System.out.println(message);
         nPrintln_v1(message, times - 1); } //Base case times == 0
   }

   //Version 2
   public static void nPrintln_v2(String message, int times)
   {  if (times == 0) //Base case
         return; //Do nothing, quit
      else {
         System.out.println(message);
         nPrintln_v2(message, times - 1); } //Recursive call
   }
}
```

Example: Sum Series *i/(2i+1)*

1/3 + 2/5 + 3/7 + 4/9 + 5/11 + ... + i/(2i+1)

```
//Sum series 1/3 + 2/5 + 3/7 + 4/9 + 5/11 + ... + i/(2i+1)
public class SumSeries
{
   public static void main(String[] args)
   {
     //print Sum series for i=3, which is 1.1619047
     double sum_3 = Sum(3);
     System.out.println(sum_3);
   }

   public static double Sum(int i)
   {
     if (i == 1) //Base case
        return 1.0 / 3;
     else
        return Sum(i-1) + (i*1.0)/(2*i + 1); //Recursive call
   }
}
```

Example: GCD (a,b)

```
//Find the GCD of 2 positive integers
import java.util.Scanner;
public class RecursiveGCD
{
  public static void main(String[] args) {
   Scanner input = new Scanner(System.in);
   System.out.print("Enter first positive number: ");
   int a = input.nextInt();
   System.out.print("Enter second positive number: ");
   int b = input.nextInt();
   System.out.println(" GCD of " + a + " and " + b + " is " + gcd(a,b));
  }

  public static int gcd(int a, int b) {
   int remainder = a % b;
   if (remainder == 0) //Base case
      return b;
   else
      return gcd(b, remainder); //Recursive call
  }
}
```

Example: Return the Reverse of a a Word

```
//Reverse a word
import java.util.Scanner;
public class ReverseWord
{
  public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    System.out.print("Enter a word: ");
    String word = input.nextLine();
    String reverse = ""; //initially empty
    reverse = reverseWord(word, reverse);
    System.out.print("The reverse of " + word + " is " + reverse);
  }

  // From left to right, take one letter at a time and add
  // to reverse from right to left
  public static String reverseWord(String word, String reverse) {
    if(word.equals("")) //Base case
       return reverse;
    else
       return reverseWord(word.substring(1, word.length()),
                        word.charAt(0) + reverse); //Recursive call
   }
}
```

End of Slides