

MODULE 5 - SHEET 1

```
public class RecFrac
{ public static void main(String[] args)
  { for (int i=0; i<=10; i++)
    System.out.println(f(i));
  }

  private static double f(int n)
  { if (n == 0)
    return 2.0d;
    else
    return 1.0d + 1.0d/(1.0d+f(n-1));
  }
}

// The body of method f may be written more economically:
//
// { return n==0 ? 2.0d : 1.0d+1.0d/(1.0d+f(n-1));
// }
```

```
public class Factorial
{ public static void main(String[] args)
  { for (int i=0; i<=10; i++)
    System.out.println(fac(i));
  }

  private static int fac(int n)
  { return n==0 ? 1 : n*fac(n-1);
  }
}
```

```
public class Hanoi
{ public static void main(String[] args)
  { move(3, "A", "B", "C");
  }
}
```

```

private static void move(int n, String p, String q, String r)
{ if (n>0)
  { move(n-1, p, r, q);
    System.out.println("Move disc " + n +
                      " from peg " + p + " to peg " + r);
    move(n-1, q, p, r);
  }
}
}

```

MODULE 5 - SHEET 2

```
public class FactorialWR
{ public static void main(String[] args)
  { System.out.println(fac5());
  }

  private static int fac5()
  { return 5*fac4();
  }

  private static int fac4()
  { return 4*fac3();
  }

  private static int fac3()
  { return 3*fac2();
  }

  private static int fac2()
  { return 2*fac1();
  }

  private static int fac1()
  { return 1*fac0();
  }

  private static int fac0()
  { return 1;
  }
}
```

```
public class HanoiWR
{ public static void main(String[] args)
```

```

    { move3("A", "B", "C");
    }

private static void move3(String p, String q, String r)
{ move2(p, r, q);
  System.out.println("Move disc 3" +
                    " from peg " + p + " to peg " + r);
  move2(q, p, r);
}

private static void move2(String p, String q, String r)
{ move1(p, r, q);
  System.out.println("Move disc 2" +
                    " from peg " + p + " to peg " + r);
  move1(q, p, r);
}

private static void move1(String p, String q, String r)
{ move0(p, r, q);
  System.out.println("Move disc 1" +
                    " from peg " + p + " to peg " + r);
  move0(q, p, r);
}

private static void move0(String p, String q, String r)
{
}
}

```

MODULE 5 - SHEET 3

// The following is a complete Java program. The problem is to analyse
// the program and determine what it writes out WITHOUT keying the program
// in and running it.

```
class K
{ public static int k = 0;
}

abstract class JJ
{ public abstract void upk();
}

class Jack extends JJ
{ public void upk()
  { K.k += 10;
  }
}

class Jill extends JJ
{ public void upk()
  { Jack ink = new Jack();
    ink.upk();
    K.k += 200;
  }
}

public class UpkEtc
{ public static void main(String[] args)
  { Jack ink = new Jack();
    fred(ink, 3000);
  }
}
```

```

        System.out.println("Value is " + K.k);
    }

    private static void fred(JJ uk, int n)
    { int a = 10*n;
      uk.upk();
      if (n > 1000)
      { K.k++;
        Jill ink = new Jill();
        fred(ink, n-1000);
        a += n;
      }
      K.k += a;
    }
}

```