Chapter Five: Thinking Abstractly (Decimals)

double

This is a new data type. Up to now we have only been dealing with whole numbers. Any variable we have used must be declared before we use it.

```
   int n;   //n is a variable to contain an integer number
```

Now we introduce a new data type – "double". This can be used to represent a decimal number.

```
public static void main(String[] args)
{
   double a = IBIO.inputDouble("enter first number  ");
   double b = IBIO.inputDouble("enter second number ");
   double num = a / b;

   IBIO.output("division gives " + num);
}
```

"output" will print the number. You can see that the accuracy is very high.

Sequence

The next program will add the numbers together 1 + 1/3 + 1/9 + 1/27 + 1/81 + … It will add 100 of these together. Notice important things about this program. "i" must be declared as an integer as it is used in a "for" statement. The variable "sum" and "term" have both been given descriptive names. In the program we use each term to calculate the next term.

```
public static void main(String[] args)
{
   double   term = 1;
   double   sum  = 0;

   for (int i = 0; i < 100; i++)
   {  sum = sum + term;
      term = term / 3;
   }

   IBIO.output("total is " + sum);
}
```

Pr 5.1 Write a program that will add up the sequence
1/5 + 1/25 + 1/125 + 1/625 + … for 100 terms. (0.25)

In the example above we used one term to create the next term. For some sequences this is difficult and it is best to create each new term from scratch as in the next example.

Pr 5.2 Write a program that it adds up the sequence
1/1 + 1/4 + 1/9 + 1/16 + 1/25 + … for 100 terms (1.6348839001848923)

Alternating Sequence

If we have an alternating sequence then the problem is more complicated. An alternating sequence is one that the terms alternately add then subtract.

```
double   term = 1;
for (int i = 1 ; i < 10 ; i++)
{  IBIO.output(term);
   term = term + 3;
}
```

This creates a sequence of numbers. 1 , 4 , 7 , 10 , 13 ,…

```
double   term = 1;
int      sign = 1;
for (i = 1 ; i < 10 ; i++)
{  IBIO.output(sign * term);
   term = term + 3;
   sign = sign * -1;
}
```

This creates a sequence of numbers. 1 , -4 , 7 , -10 , 13

Pi

Pi can be calculated using the formula:

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \ldots$$

Pr 5.3 Write a program that will add up the sequence discussed above to 10,000 terms. Output 4 times the answer to get pi. (3.1414926535900345)

$$\frac{\pi}{2} = 1 + \frac{1}{3} + \frac{1 \times 2}{3 \times 5} + \frac{1 \times 2 \times 3}{3 \times 5 \times 7} + \frac{1 \times 2 \times 3 \times 4}{3 \times 5 \times 7 \times 9} + \ldots$$

Pr 5.4 The sequence above is a much quicker way of calculating pi Write a program that will add up the sequence above to 100 terms. Output double the answer (3.1415926535897922)