Practical Exercise 7 – For loops

The purpose of this task is to provide an introduction to *for* loops.

The following code provides two examples of how *for* loops can be used to display a list of values.

Notice the two different approaches to setting the y-coordinate of the output. What are the advantages and limitations of each approach?

**public** **void** paint(Graphics g)

{

**for**(**int** i=0; i<10; i++)

{

g.drawString(""+i, 20, i\*20+20);

}

int yPos=20;

**for**(**char** ch='a'; ch<='z'; ch++)

{

g.drawString(""+ch, 100, yPos);

yPos=yPos+20;

}

}

# Task 1 – FOUR for loops

1. Modify the sample code to output each of the following sequences using a for loop in a single window.

|  |  |  |  |
| --- | --- | --- | --- |
| 2  4  6  8  10  12  14  16  18  20 | 10  9  8  7  6  5  4  3  2  1  blast off !!!! | Group 10 :  Group 20 :  Group 30 :  Group 40 :  Group 50 : | 1 \* 5 = 5  2 \* 5 = 10  3 \* 5 = 15  4 \* 5 = 20  5 \* 5 = 25  6 \* 5 = 30  7 \* 5 = 35  8 \* 5 = 40  9 \* 5 = 45  10 \* 5 = 50  11 \* 5 = 55  12 \* 5 = 60 |

1. Alter the last program above (Part 1d, that prints out the 5 times table) so that the user can enter the number for the particular times table they want.

# Task 2 – Vowels and squares

Write a program that contains two *for loops* that generate the following lists.

1. The first displays the letters of the alphabet, with all vowels replaced with the word “vowel”.
2. The second displays the integers from 1 to 25 and determines if the number is a square.

The output should be something like this…

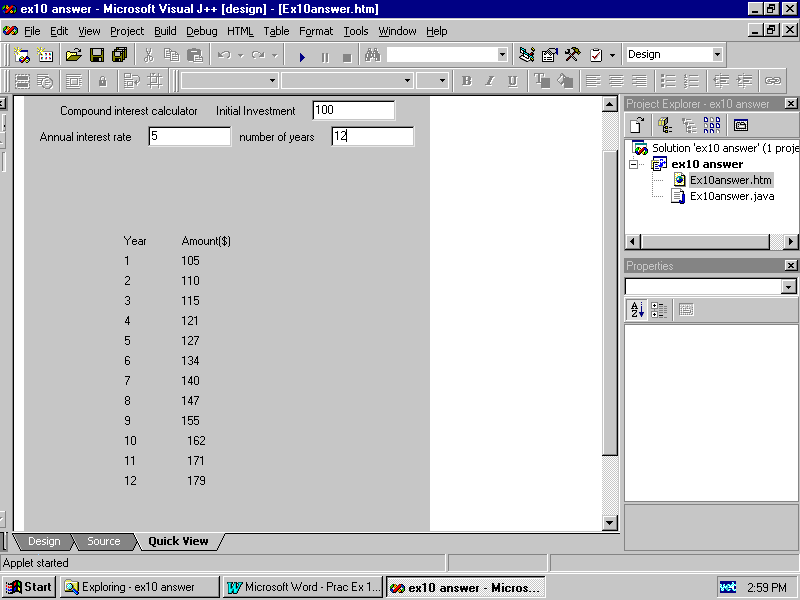
|  |  |
| --- | --- |
| vowel  b  c  d  vowel  f  g  h  vowel  j  *… and so on ...*  x  y  z | 1  2  3  4 is 2 squared  5  6  7  8  9 is 3 squared  10  *… and so on …*  25 is 5 squared |

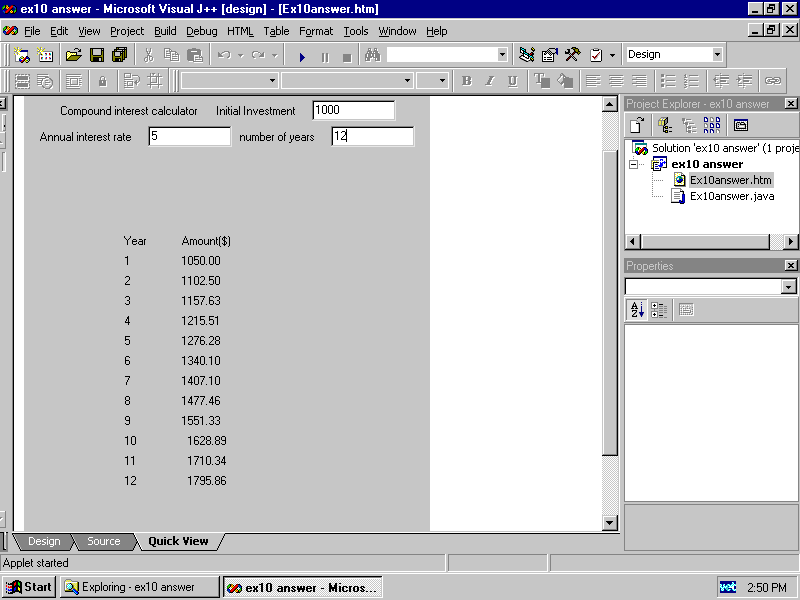
Hint: use a *switch* or an *if* statement in a for loop.

# Task 3 – The Miracle of Compound Interest

1. Write a program to perform a simple compound interest calculation.  
     
   The user should be asked to enter:   
     
    principal initial amount invested  
    rate (r) annual interest rate  
    time (t) number of years  
     
   The program should produce a table as per the first example below showing the value of the investment for each year up to the length of the investment.   
     
   If the time of the investment is less than 1 year, an error should be printed.  
     
   The formula is: amount = principal \* (1 + rate/100 )^time

in Java amount = principal \* Math.pow(1.0 + rate/100.0, time);





1. Alter your program so that it looks like the second example. i.e. the amount is printed out with two decimal places. To do this you must add four lines to your program in the appropriate places:

import java.text.DecimalFormat;

DecimalFormat precision2; //as a global variable

precision2 = new DecimalFormat(“#0.00”); // in the init() method and

precision2.format(amount) // within your g.drawString statement

# Task 4 – Another brick in the wall (optional bonus task for fun)

## Produce a brick pattern effect using

g.drawRect(x, y, width, height)

Hints: This problem requires a nested loop.

Start by producing a single row first.