IRA EXAMPLES

This topic has two examples showing the calculation of the future value an IRA (Individual Retirement Account).

Definite Counting Loop Example – IRA After \( x \) Years

This first example illustrates how to combine the suggestions covered in the topic Help for Beginners. Specifically, we’re going to follow the technique How to Invent an Algorithm (version 2) but as we do so, keep an eye out for where we can use the definite counting loop idiom.

Pretend you’ve won $40,000 in the state lottery. You invest it in an IRA that pays a fixed 5% APR (annual percentage rate) compounded monthly. You want to know what its value will be in 20 years. Write a Java application that finds out.

How to Invent an Algorithm (version 2)

<table>
<thead>
<tr>
<th>Step</th>
<th>What?</th>
<th>How?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Understand the problem</td>
<td>• Use pencil and paper to solve the problem by hand.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Use all the problem solving techniques available.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Write a list of any questions &amp;or ambiguities you</td>
</tr>
<tr>
<td></td>
<td></td>
<td>encounter and seek answers.</td>
</tr>
</tbody>
</table>

To calculate the amount of interest for one month, multiply the $40,000 by the monthly interest rate, which is \( \frac{5}{12} \) the annual rate. “5%” means “5 out of 100.” Therefore, the computation of the first month’s interest is:

\[
$40,000 \times \frac{5}{100} \times \frac{1}{12} = $166.67
\]

“Compounded monthly” means that you add the month’s interest onto the IRA balance so that during the following month you earn interest on the total balance that includes the first month’s interest. Thus, the second month’s interest is:

\[
$40,166.67 \times \frac{5}{100} \times \frac{1}{12} = $167.36
\]
Continuing, I calculate the interest and balance for several years.

<table>
<thead>
<tr>
<th>Month</th>
<th>IRA Balance (before)</th>
<th>Multiplied By</th>
<th>Interest</th>
<th>IRA Balance (after)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$40,000.00</td>
<td>5/100/12</td>
<td>$166.67</td>
<td>$40,166.67</td>
</tr>
<tr>
<td>2</td>
<td>40,166.67</td>
<td>5/100/12</td>
<td>167.36</td>
<td>40,334.03</td>
</tr>
<tr>
<td>3</td>
<td>40,334.03</td>
<td>5/100/12</td>
<td>168.06</td>
<td>40,502.09</td>
</tr>
<tr>
<td>4</td>
<td>40,502.09</td>
<td>5/100/12</td>
<td>168.76</td>
<td>40,670.84</td>
</tr>
</tbody>
</table>

At this point, I see that I can get the final answer by doing this computation for 20×12 months (or 20 years). I stopped the pencil and paper computation because if I did it for all 20 years there’d be no need to write a Java program. On the other hand, what good is a program that works only for such specific data? For example, what if you can’t get 5% APR?

So, let’s continue algorithm development by opening up the problem.

<table>
<thead>
<tr>
<th>Step</th>
<th>What?</th>
<th>How?</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Determine the program’s requirements</td>
<td>• Identify the program’s output.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Work backwards from the output to determine what input values are necessary to produce it.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Determine what computations transform the input into the output.</td>
</tr>
</tbody>
</table>

• **input** the starting value of the IRA, the annual percentage rate and the number of years to carry on the calculation
• **calculate** the value of the IRA each year for the given number of years at the given APR compounded monthly
• **output** the final value of the IRA
<table>
<thead>
<tr>
<th>Step</th>
<th>What?</th>
<th>How?</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Write down a first-draft algorithm</td>
<td>• Write in English the steps you employed when solving the problem by hand. Don’t worry about language or details; just get your ideas on paper in outline form.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Returning to my pencil and paper work in step 1, I now write the steps that I employed:

```
input starting value 40,000, apr of 5% and number of years 20
During month 1, 40,000 × 5/100/12 = 166.67
new IRA value is 40,000 + 166.67 = 40,166.67
During month 2, 40,166.67 × 5/100/12 = 167.36
new IRA value is 40,166.67 + 167.36 = 40,334.03
ETC.
```

<table>
<thead>
<tr>
<th>Step</th>
<th>What?</th>
<th>How?</th>
</tr>
</thead>
</table>
| 4   | Refine your algorithm | • Identify loops.  
  ○ In your draft outline, look for (1) verbs that imply repetition (e.g. repeat) and (2) repetitive sentences.  
  ○ Isolate these sentences and wrap them within an appropriate looping statement. |
To identify the loop, I see that steps 2 and 3 in my paper and pencil work are repeated for each month. Step 2 uses the current value of the IRA to calculate the interest and step 3 uses the current value of the IRA and the interest to compute the IRA value to be used in the next loop cycle:

input starting value 40,000, apr of 5% and number of years 20

During month 1, 40,000 × 5/100/12 = 166.67

new IRA value is 40,000 + 166.67 = 40,166.67

During month 2, 40,166.67 × 5/100/12 = 167.36

new IRA value is 40,166.67 + 167.36 = 40,334.03

---

<table>
<thead>
<tr>
<th>Step</th>
<th>What?</th>
<th>How?</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Refine your algorithm</td>
<td>• Introduce program variables suggested by the nouns in your outline.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Noun</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>value of IRA</td>
<td>iravalue</td>
</tr>
<tr>
<td>annual percentage rate</td>
<td>apr</td>
</tr>
<tr>
<td>number of years to calculate</td>
<td>years</td>
</tr>
<tr>
<td>month counter</td>
<td>month</td>
</tr>
<tr>
<td>amount of month's interest</td>
<td>interest</td>
</tr>
<tr>
<td>Step</td>
<td>What?</td>
</tr>
<tr>
<td>------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>4</td>
<td>Refine your algorithm</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The loop that I identified in the paper and pencil work above is a definite counting loop using `month` as the loop control variable.

Here’s the first draft of the pseudo-code.

```plaintext
input iraValue, apr, years
month = 0;
while ( )
{
    ...
    month++;
}
print iraValue
```

The loop must quit after the last month of the last year.

```plaintext
input iraValue, apr, years
month = 0;
while ( month < years * 12 )
{
    ...
    month++;
}
print iraValue
```
Take the repeated statements out of the pencil and paper work shown on page 5, alter the nouns to variable names and use some programming notation to express the calculations and we get this final pseudo-code algorithm:

```
Algorithm

input iraValue, apr, years
month = 0;
while ( month < years * 12 )
{
    interest = iraValue*apr/100/12;
    iraValue += interest;
    month++;
}
print iraValue
```

The coded Java application is shown on the next page.
import java.text.DecimalFormat;
import java.util.Scanner;
import static javax.swing.JOptionPane.*

public class IRA1
{
    public static void main( String [] args )
    {
        // declare data
        double iraValue;  // current value of the IRA
        double apr;       // annual percentage rate
        int years;        // number of years to calculate
        // input these values
        String prompt, input, output;
        prompt = "Enter initial IRA deposit, APR "
        + "and number of years to calculate";
        input = showInputDialog( prompt );
        Scanner in = new Scanner( input );
        iraValue = in.nextDouble( );
        apr = in.nextDouble( );
        years = in.nextInt( );
        // iterate over the number of years
        // compounding is monthly
        int month = 0;
        while ( month < years*12 )
        {
            double interest = iraValue * apr/100./12.;
            iraValue += interest;
            month++; // count the passing month
        }
        // print results
        DecimalFormat df = new DecimalFormat( "$#,###.00" );
        output = "IRA value after " + years + " years is "
        + df.format( iraValue );
        showMessageDialog( null, output );
    }
}
**Indefinite Counting Loop Example – How Many Years for IRA**

Suggestion #2 in *Help for Beginners* advises you, when faced with a programming problem, to look for solutions to similar problems. This example illustrates this technique by solving a variation of the previous example.

Having invested $40,000 in an IRA, you want to know how many years it will take to grow it into $1 million. Again, assume 5% APR compounded monthly.

This program requires the same calculations as the previous example. The difference is that an indefinite loop is called for that quits when the IRA’s value reaches $1 million. We take and edit the algorithm for the previous example:

```plaintext
input iraValue, apr, years
month = 0;
while (month < years * 12)
{
    interest = iraValue*apr/100/12;
    iraValue += interest;
    month++;
}
print iraValue

Quit when IRA value reaches $1 million
Calculate number of years as month/12
with extra months being month % 12
```
Rewritten into pseudo-code:

```
rewritten into pseudo-code:

input iraValue, apr
month = 0;
while ( iraValue < 1 million )
{
    interest = iraValue*apr/100/12;
    iraValue += interest;
    month++;
}
years = month / 12;
print years, month % 12
```

The algorithm coded into Java is shown on the next page.
```java
import java.text.DecimalFormat;
import java.util.Scanner;
import static javax.swing.JOptionPane.*;

public class IRA2
{
    public static void main( String [] args )
    {
        // declare data
        double iraValue;  // current value of the IRA
        double apr;       // annual percentage rate
        int years;        // number of years

        // input these values
        String prompt, input, output;
        prompt = "Enter initial IRA deposit and APR ";
        input = showInputDialog( prompt );
        Scanner in = new Scanner( input );
        iraValue = in.nextDouble( );
        apr = in.nextDouble( );

        // iterate until IRA value reaches $1 million
        // compounding is monthly
        int month = 0;
        while ( iraValue < 1E6 )
        {
            double interest = iraValue * apr/100./12.;
            iraValue += interest;
            month++; // count the passing month
        }

        // print results
        years = month / 12;
        int extra = month % 12;
        DecimalFormat df = new DecimalFormat( "$#,###.00" );
        output = "IRA value will reach " + df.format( iraValue )
            + " after " + years + " and " + extra + "months";
        showMessageDialog( null, output );
    }
}
```
## Programming Exercises

1. Write a Java application that estimates and prints the mathematical constant $e$ (equal to 2.71828). You can estimate its value using the formula:

\[
e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \ldots + \frac{1}{k!} + \ldots
\]

Quit looping when the term $\frac{1}{k!}$ becomes less than $10^{-4}$.

2. Write a Java application that inputs the number of ants in your ant farm and calculates the number of months it will take for the ant population to double if it increases by 2% each month.

3. Strontium-90 decays at a rate of 2.4% per year. Write a Java application to calculate and print its half-life (the number of years it takes to degrade into half its amount). For your output print:

\[\text{Strontium-90 half-life is between } x \text{ and } x+1 \text{ years}\]

For example, if its half-life is 8.25 years then print 8 for $x$ and 9 for $x+1$.

4. Flatland has a population of 165,000, 58% of which are Squares and the rest Circles. The number of Squares increases by 1.5% per year; the number of Circles by 2%. Circle leaders want to know how many years it will take Circles to outnumber Squares so they can win all the elections. Write a Java application that gives them the answer.

5. Pretend that Grandma lives in a nursing home that charges $5,000 per month. She receives $2,000 per month in Social Security payments. The remaining charges must come out of her savings, which amounts to $100,000 invested at 5% APR compounded monthly.

When Grandma's savings are reduced to $1,000, Medicaid will take over her nursing home payments. How many years and months will that be? Write a Java application to find out.

6. Suppose you open an IRA with $5,000. The IRA pays a fixed 5% APR compounded monthly. Every year you add $5,000 to it. You want to know how long it will take to grow to $1 million. Write a Java application that finds out and prints the number of years and months.