

Installation Instruction for Corejava package:

For Windows XP/Vista

When you unzip the file called CoreJavaBook.zip you will find two subdirectories

- api

This directory has all documentation for the functions that come with this package. You can use index.html to access these instructions.

- corejava

This directory has source files. (.java Files)

You must unzip both of these directories into a directory called CoreJavaBook.

Add the directory to your CLASSPATH as follows:

On Windows -

You can go Control Panels --> System --> Click on Advanced Tab --> Environment Variables and then check to see if CLASSPATH variable exists. If so, then update it otherwise add a new environment variable.

Updating existing values -

Let's say your existing CLASSPATH variable value is set to -

".;C:\Program Files\Java\jre1.6.0_02\lib\ext\QTJava.zip"

then you should change this value to

".;C:\CoreJavaBook;C:\Program Files\Java\jre1.6.0_02\lib\ext\QTJava.zip"

Adding a new value under User Variables or System Variables

Click on new and set the values as follows:

Variable name: CLASSPATH

Variable Value: .;C:\CoreJavaBook

Make sure there are no extra spaces at the end of each string. This can often lead to errors when compiling your java programs.

To use CoreJava Package in the CIS Lab?

Unzip the CoreJavaBook directory in the Z drive.

Make sure the CoreJavaBook directory is a root directory.

In lab you can setup a batch file and set up the classpath manually by running the batch file.

```
set CLASSPATH=.;H:\CoreJavaBook
```

```
echo %CLASSPATH%
```

Save this file in set.bat in the Z drive.

Now you can run the batch file by using set

On Unix

Using the CLASSPATH environment variable

In general, you will want to use the `-classpath` command-line option, as explained in the previous section. This section shows you how to set the CLASSPATH environment variable if you want to do that, or clear settings left over from a previous installation.

Setting CLASSPATH

In `csh`, the CLASSPATH environment variable is modified with the `setenv` command. The format is:

```
setenv CLASSPATH path1:path2
```

In `sh`, the CLASSPATH environment variable can be modified with these commands:

```
CLASSPATH = path1:path2:...  
export CLASSPATH
```

For example -

```
CLASSPATH=.:$HOME/CoreJavaBook  
export PATH  
export CLASSPATH
```

Clearing CLASSPATH

If your CLASSPATH environment variable has been set to a value that is not correct, or if your startup file or script is setting an incorrect path, you can unset CLASSPATH in `csh` by using:

```
unsetenv CLASSPATH
```

In `sh`, you would use:

```
unset CLASSPATH
```

These commands unset CLASSPATH for the current shell only. You should also delete or modify your startup settings to ensure that you have the right CLASSPATH settings in future sessions.

Changing Startup Settings

If the CLASSPATH variable is set at system startup, the place to look for it depends on the shell you are running:

Shell	Startup Script
-------	----------------

<code>csh</code> , <code>tcsh</code>	Examine your <code>.cshrc</code> file for the <code>setenv</code> command.
--------------------------------------	--

<code>sh</code> , <code>ksh</code>	Examine your <code>.profile</code> file for the <code>export</code> command.
------------------------------------	--

Example of how to use this package

Here is an example to get you started with this package:

```
import corejava.*; //imports the package, giving access to all methods for I/O
import java.text.*; //java.lang package for Input/Output

public class Mortgage
{
    public static void main(String[] args)
    {
        double principal;
        double yearlyInterest;
        int years;
        principal = Console.readDouble("Loan amount (no commas:");
        //reads a double
        yearlyInterest = Console.readDouble ("Interest rate in % (ex: use 7.5 for
        7.5%):")/100;
        //reads a double and divides it by 100
        years = Console.readInt("The number of years:"); //reads a Int
        //Similarly you can use readLine to read a String.

        double monthlyInterest = yearlyInterest / 12;
        double payment = principal * monthlyInterest / (1 - (Math.pow(1/(1 +
        monthlyInterest), years * 12)));
        System.out.println("Your payment is ");
        NumberFormat nf = NumberFormat.getCurrencyInstance();
        //Creates an object for formatting data
        System.out.println(nf.format(payment)); //Formats the data
    }
}
```