

Forum:[Hack Forums](#)

Thread: [Decimal to Binary Converter](#)

CyberPunkw - March 28, 2015, 2:44 pm

Here is a decimal to binary converter I made. It was inspired by a program in a java textbook I have.

```
Code public class LocalClassDemo
{
    public static void main(String[] args)
    {
        class ShowBits
        {
            int numbits;

            ShowBits(int n)
            {
                numbits = n;
            }

            void show(long val)
            {
                long mask = 1;

                mask <=<= numbits-1;

                int spacer = 0;
                for (; mask != 0; mask >>= 1)
                {
                    if ((val & mask) !=0) System.out.print("1");
                    else System.out.print("0");
                    spacer++;
                    if ((spacer % 8) == 0)
                    {
                        System.out.print(" ");
                        spacer = 0;
                    }
                }
                System.out.println();
            }
        }

        long userInput = Long.parseLong(args[0]);

        ShowBits byteval = new ShowBits(64); // 64 is the max bits long can handle
        System.out.println(userInput + " in binary");
        byteval.show(userInput);
    }
}
```

Barinade - March 28, 2015, 8:00 pm

```
Code    public static String getBinary(int i) {  
        String binary = "";  
        if (i > 0) {  
            binary += getBinary(i/2) + (i % 2);  
        }  
        return binary;  
    }
```

}Why are you declaring ShowBits in your main method? Never seen that before

Psifour - March 28, 2015, 8:00 pm

Did you just define an internal class within a static method? I almost have to applaud you for making me want to cry.

Other than that (and yes I understand why you did it) this doesn't look too bad (still plenty of ways to both optimize and improve this though).

(Mar 28, 2015 08:00 PM)Barinade Wrote: [\[CITE\]](#) Code public static String getBinary(int i) {
 String binary = "";
 if (i > 0) {
 binary += getBinary(i/2) + (i % 2);
 }
 return binary;
 }

}Why are you declaring ShowBits in your main method? Never seen that before

I typically avoid recursion like a plague, but this is a neat way to go about this. My only complaint is the 128 String concatenations that would happen with a long, but that is a minor issue and could be resolved without too much effort.

Barinade - March 28, 2015, 8:11 pm

(Mar 28, 2015 08:00 PM)Psifour Wrote: [\[CITE\]](#) Did you just define an internal class within a static method? I almost have to applaud you for making me want to cry.

Other than that (and yes I understand why you did it) this doesn't look too bad (still plenty of ways to both optimize and improve this though).

I typically avoid recursion like a plague, but this is a neat way to go about this. My only complaint is the 128 String concatenations that would happen with a long, but that is a minor issue and could be resolved without too much effort.

Everything here is just for learning purposes, he wanted to do it his way to learn, I've been working with recursion so I recreated the method accordingly

Actual implementation of the method is, if I'm using the word correctly, redundant simply because we have:

```
Integer.toBinaryString(int)
```

Psifour - March 28, 2015, 8:17 pm

(Mar 28, 2015 08:11 PM)Barinade Wrote: [\[CITE\]](#) Everything here is just for learning purposes, he wanted to do it his way to learn, I've been working with recursion so I recreated the method accordingly

Actual implementation of the method is, if I'm using the word correctly, redundant simply because we have:

```
Integer.toBinaryString(int)
```

Understandable and agreeable, but that said I am for over-engineering my solutions when they are for practice/experimentation. It will subtly increase your ability to rapidly prototype solutions that many developers would struggle to reach as a final product even.

Barinade - March 28, 2015, 8:27 pm

(Mar 28, 2015 08:17 PM)Psifour Wrote: [\[CITE\]](#) Understandable and agreeable, but that said I am for over-engineering my solutions when they are for practice/experimentation. It will subtly increase your ability to rapidly prototype solutions that many developers would struggle to reach as a final product even.

A lot of that comes down to designing equations and understanding logistics, then there's threading which puts a lot of people behind, especially when they don't include the possibilities in their equations/logics

Recursion isn't something every gets right away because to understand recursion you must first understand recursion...

It comes down to: Is your changes between test runs based on logic or random guesses until it works?

Our goal is to get to the logic answer, and once we do, we won't need as many test runs as others.

Psifour - March 28, 2015, 8:42 pm

(Mar 28, 2015 08:27 PM)Barinade Wrote: [\[CITE\]](#) A lot of that comes down to designing equations and understanding logistics, then there's threading which puts a lot of people behind, especially when they don't include the possibilities in their equations/logics

Recursion isn't something every gets right away because to understand recursion you must first understand recursion...

It comes down to: Is your changes between test runs based on logic or random guesses until it works?

Our goal is to get to the logic answer, and once we do, we won't need as many test runs as others.

My design process is multi-stage, but I feel like my results tend to be better than average. I start by creating three concepts that have no critical flaws in their design/logic (these are not prototyped at this point and if I can deduce a potential flaw they are replaced with a better concept). After that I compare the memory, processing, and other resource requirements for each of the three and weight that based on the target environment. If the code will be used as a library or potentially has use in other projects I then factor in the flexibility of each method to facilitate code-reuse. This typically leaves me with a clear winner, but regardless all remaining concepts that are within a reasonable margin of being the best (rarely all three) get a rough version coded and benchmarked in the target environment.

After that it is just a matter of iterative design on the best performing method.

That said I do occasionally "slack it up" on personal projects and implement poorly designed options simply due to the reduction in development time they provide (I know shame on me).

CyberPunkw - March 30, 2015, 4:09 pm

(Mar 28, 2015 08:00 PM)Barinade Wrote: [\[CITE\]](#) Why are you declaring ShowBits in your main method? Never seen that beforeWell, I realized that you can, so I decided to just try It. I'm new to Java (started about 3 weeks ago) so I am at the stage where I am just trying things out.

Psifour - March 30, 2015, 6:01 pm

(Mar 30, 2015 04:09 PM)CyberPunkw Wrote: [\[CITE\]](#) Well, I realized that you can, so I decided to just

try It. I'm new to Java (started about 3 weeks ago) so I am at the stage where I am just trying things out.

I do see the potential to produce some super silly code. That said it really is terrible form :P.

CyberPunkw - *March 30, 2015, 10:18 pm*

(Mar 30, 2015 06:01 PM)Psifour Wrote: [\[CITE\]](#) I do see the potential to produce some super silly code. That said it really is terrible form :P.

Well, its not exactly production code now is it?