22MCA0139
RAJAT SINGH

Software Process Metrics

Digital Assignment

⊛ Code

```
public class ArmstrongAndFibonacci{
    public static void main(String[] args){
        int lowerBaund = 1;
        int upperBand  = 1000;
        System.out.println("Armstrong Numbers:\n");
        for (int num = lowerBaund; num <= upperBaund; num++){
            if (isArmstrongNumber(num)){
                System.out.println(num + " ");
            }
        }
        System.out.println("Fibonacci Numbers:\n");
        int prev = 0, curr = 1;
        while(curr <= upperBaund){
            if(curr >= lowerBaund){
                System.out.println(curr + " ");
            }
            int neat = prev + curr;
            prev = curr;
            curr = neat;
        }
    }
public static boolean isArmStrangNumber(int num) {

    int n = num;
    int digit = string.valueOf(num).length();
    int sum = 0;
    while (num > 0) {

        int dig = num % 10;
        sum += Math.pow(dig, digit);
        num /= 10;
    }
    return sum == n;
  }
}
```

---

1) Lines Of Code (LOC)

LOC = non-commented source line + commented source line

= NLOC + CLOC

= 33 + 0 = 33 lines

2) Halstead Metrics

Operators	Occurences	Operands	Occurrences
int	8	++	1
{}	8	>	1
()	15	%	1
,	1	+=	1
=	12	/=	1
<=	2	==	1
>=	1	if	2
+	3	for	1
;	17	return	1
lowerBand	3	while	2
num	11	upperBand	3
curr	7	prev	3
n	2	next	2
sum	3	digit	2
		dig	2

Vocabulary, $n = n_1 + n_2 = 29$

Size, $N = N_1 + N_2 = 117$

$n_1 = 19$ $\quad N_1 = 79$

Volume (V) $= N \log_2 (n) = 117 \log_2 (29) = 568.384$

Level (L) $= \frac{2}{n_1} \times \frac{n_2}{N_2} = \frac{2}{19} \times \frac{10}{38} = 0.027$

Difficulty (D) $= \frac{n_1}{2} \times \frac{N_2}{n_2} = 75.05$

$$\text{Effort}(E) = \frac{V}{L} = \frac{568.384}{0.027} = 21051.2592$$

$$\text{Faults } B = \frac{V}{S^*} = \frac{568.384}{3000} = 0.1894$$

$$\text{Testing Time }(T) = \text{Effort}/S = \frac{21051.2592}{20}$$

$$= 1052.56296$$

3) Functional Point

External Query (EQ) = 0

External Interface (EIF) = 0

Internal Logic File (ILF) = 0

External Inputs (EI) = 2

External Outputs (EO) = 2

$$UFC = 3EI + 4EO + 3EQ + 7ILF + 5EIF$$

$$= 3\times2 + 4\times2 + 0 + 0 + 0 = 14$$

For Technical Complexity Factors,

Technical Factors	Weights	Scale Values (0-5)
Complexity of Algorithms	3	4
Reusability	3	5
Error Handling	2	2
Operational Ease	1	4
Documentation	1	0
Turn Around Time	3	5

$$TCF = (3\times4) + (3\times5) + (2\times2) + (1\times4) + (1\times0) + (3\times5) = 12 + 15 + 4 + 4 + 0 + 15$$

$$= 50$$

Funchional Point(FP) $= UFC \times (0.65 + \cancel{\times}(0.01 \times TCF))$

$$= 14 \times (0.65 + (0.01 \times 50))$$

$$= 16.1 \simeq 16$$

4) Object Point

$$OP = FP \times LOC$$

$$= 16 \times 25 = 400$$

Let us consider a ~~database~~ database application project with four screens and four views each and seven data tables for three servers and four clients. The application may generate two reports of six sections each from seven data tables for two servers and three clients.

For Screens,

No. of Screens = 4

For each screen,

no of views = 4

" " data tables = 7

" " servers = 3

" " clients = 4

From the information above we consider the complexity of each screen = Medium (3) for Reports,

No. of rewards = 2
" " sections = 6
" " data tables = 7
" " servers = 2
" " clients = 3

from the information above we consider the complexity to be = Difficult (6)

Now,

Object Points = Σ [no. of object instances * complexity weight of each interface]

$= 4 \times 3 + 2 \times 6 = 12 + 12 = 24$

∴ There are 24 Object Points.

3/8/23