

# UML DIAGRAM

Reverse

Number: int

+ Reverse()

+ reverseDigit(int): int

+ reverseString(string): string

Fibonacci

+ Fibonacci()

+ return Fibonacci(int): int

EfficientFibonacci

+ EfficientFibonacci()

+ return EfficientFibonacci(int): int

Description:

Reverse

\* Reverse() → Construction of class Reverse and initializes (int number) the integer in class Reverse

\* reverseDigit(int) → The function takes in a non-negative integer value and reverse the digit's using recursion. The reversed integer is then returned.

\* reverseString(string) → The function takes in a string and returns the reversed string.

Fibonacci

\* Fibonacci() → Construction of class Fibonacci and this constructor will be invoked when an object of class Fibonacci is created.

\* return Fibonacci(int) → The function takes in an integer and returns the  $n^{th}$  Fibonacci number ( $F_n = F_{n-1} + F_{n-2}$ ) using recursion.

EfficientFibonacci

\* EfficientFibonacci() → Construction of class EfficientFibonacci and this constructor will be invoked when an object of class EfficientFibonacci is created.

\* return EfficientFibonacci(int) → This function does't have role as return Fibonacci(int) but does it efficiently by storing calculated numbers. The recursion should not be called for a number that has already been calculated.

# Testing

Given Input

~~8~~

Output

12345 apple 6 6

-1 appa 20 3

11900 Adelaide Sa 11

54321 elppa 8 8

ERROR appa 6165 2

911 edialada ERROR 89

10

10

15

15

20

20

25

25

30

30

# Testing

Given Input

~~8~~

Output

12345 apple 6 6

-1 appa 20 3

11900 Adelaide Sa 11

54321 elppa 8 8

ERROR appa 6165 2

911 edialada ERROR 89

10

10

15

15

20

20

25

25

30

30