**Add**

Start >> Result = x+y >> return Result >> End

|  |  |  |
| --- | --- | --- |
| Path Type | Test Case Input | Test Case Output |
| visible | X = 5, y = 2 | Result = 7 |

**Multiply**

Start >> Result = x\*y >> return Result >> End

|  |  |  |
| --- | --- | --- |
| Path Type | Test Case Input | Test Case Output |
| visible | X = 2, y = 2 | Result = 4 |

**Subtract**

Start >> Result = x-y >> return Result >> End

|  |  |  |
| --- | --- | --- |
| Path Type | Test Case Input | Test Case Output |
| visible | X = 10, y = 2 | Result = 8 |

**Divide**

Start >> y == 0 >true> print (“you can’t divide by zero!”) >> raise zero division error >> end;

Start >> y==0 >false> x == 0 >true> return 0 >> end;

Start >> y==0 >false> x == 0 >false> result = x/y >> return result >> end;

|  |  |  |
| --- | --- | --- |
| Path Type | Test Case Input | Test Case Output |
| visible | X = 5, y = 0 | you can’t divide by zero! |
| visible | X = 0, y = 8 | 0 |
| visible | X = 10, y = 5 | Result = 2 |

**isExit**

Start >> next\_calculation =="no" >true> return true >> end;

Start -> next\_calculation =="no" >false> next\_calculation=="yes" >true> return false >> end;

Start -> next\_calculation =="no" >false> next\_calculation=="yes" >false> raise ValueError >> end;

|  |  |  |
| --- | --- | --- |
| Path Type | Test Case Input | Test Case Output |
| visible | next\_calculation = ‘no’ | True |
| visible | next\_calculation= ‘yes’ | False |
| visible | next\_calculation = ‘w’ | ValueError |

**Calculate**

Start >> not num1 or not num2? >true> print (inputs can not be null") >> raise valueerror >> end;

Start >> not num1 or not num2? >false> num1= check\_user\_input(num1) >> num2 = check\_user\_input(num2) >> result = "" >> choice in (1,2,3,4) >false> raise exception “Invalid choice!” >> end;

Start >> not num1 or not num2? >false> num1= check\_user\_input(num1) >> num2 = check\_user\_input(num2) >> result = "" >> choice in (1,2,3,4) (true) >> choice =="1"? >true> result = add (num1, num2) >> return result >> end;

Start >> not num1 or not num2? >false> num1= check\_user\_input(num1) >> num2 = check\_user\_input(num2) >> result = "" >> choice in (1,2,3,4) >true)> choice =="1"? >false> choice =="2"? >false> result = subtract (num1, num2) >> return result >> end;

Start >> not num1 or not num2? >false> num1= check\_user\_input(num1) >> num2 = check\_user\_input(num2) >> result = "" >> choice in (1,2,3,4) >true> choice =="1"? > no> choice =="2"? >no> choice =="3"? >yes >> result = multiply (num1, num2) >> return result >> end;

Start >> not num1 or not num2? >false> num1= check\_user\_input(num1) >> num2 = check\_user\_input(num2) >> result = "" >> choice in (1,2,3,4) >true> choice =="1"? > no> choice =="2"? >no> choice =="3"? >no> choice =="4"? > no> return result >> end;

Start >> not num1 or not num2? >false> num1= check\_user\_input(num1) >> num2 = check\_user\_input(num2) >> result = "" >> choice in (1,2,3,4) >true> choice =="1"? >no> choice =="2"? >no> choice =="3"? >no> choice =="4"? >yes> num2==0? >no> result = divide (num1, num2) >> return result >> end;

Start >> not num1 or not num2? >false> num1= check\_user\_input(num1) >> num2 = check\_user\_input(num2) >> result = "" >> choice in (1,2,3,4) >true> choice =="1"? >no> choice =="2"? >no> choice =="3"? >no> choice =="4"? >yes > num2==0? >yes> print (divide by zero) >> raise zerodivisionerror >> end;

|  |  |  |
| --- | --- | --- |
| Path Type | Test Case Input | Test Case Output |
| visible | Num1 = NULL, num2 = NULL , choice = 1 | ValueError |
| visible | num1 = 8, num2 =3, choice = 5 | Exception |
| visible | num1 = 5, num2 = 5, choice = 1 | result = 10 |
| visible | num1 = 7, num2 = 8, choice = 2 | result =-1 |
| visible | num1 = 5, num2 = 5, choice = 3 | result = 25 |
| visible | num1 = 12, num2 = 6, choice = 4 | result = 2 |
| invisible | num1 = 8, num2 = 0, choice = 5 | " " |
| visible | num1 = 6, num2 = '0’, choice = 4 | ZeroDivisionError |

**CheckUserInput**

Start >> input ==""? >yes> print ("input can’t be empty “) >> raise valueError >> end;

Start >> input ==""? >no> try >> val = int(input) >> return val >> end;

Start >> input ==""? >no> try >> catch ValueError >> try >> val = float(input) >> return val >> end;

start >> input ==""? >no> try >> catch ValueError >> try >> Catch ValueError >> print (input is not number) >> raise ValueError >> end;

|  |  |  |
| --- | --- | --- |
| Path Type | Test Case Input | Test Case Output |
| visible | input = "" | ValueError |
| visible | input = "8" | 8 |
| visible | input = "8.5" | 8.5 |
| visible | input = "w" | ValueError |