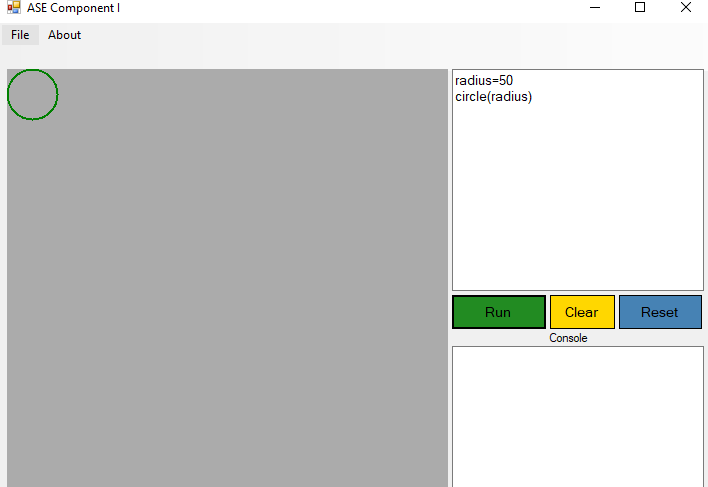
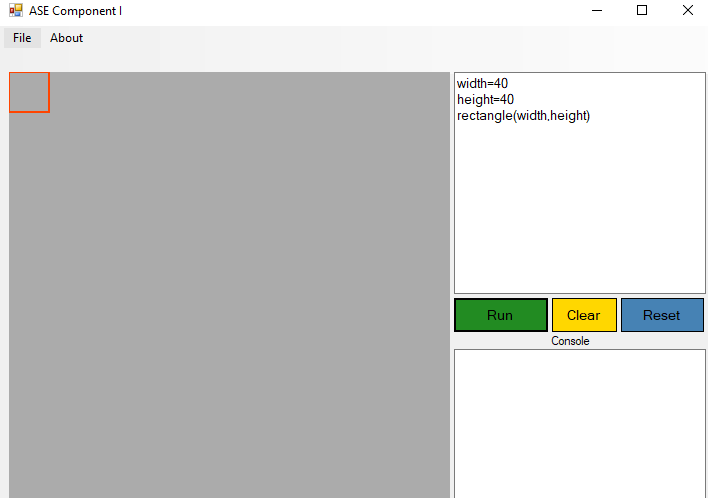
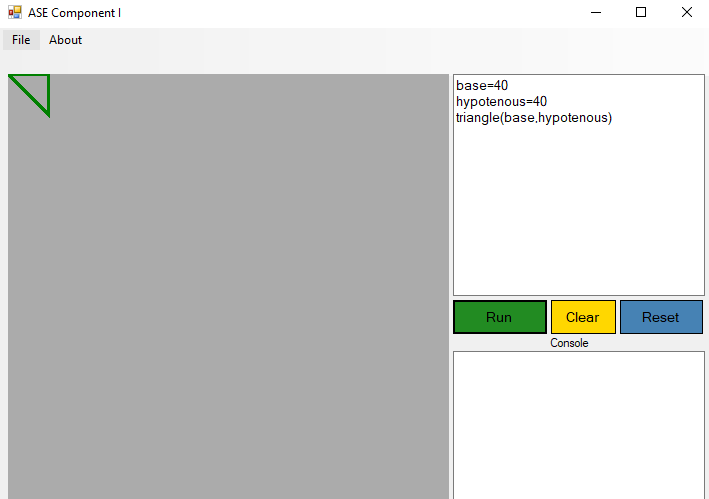
**Variables - allows variables to be used in loop and as parameters to draw commands**



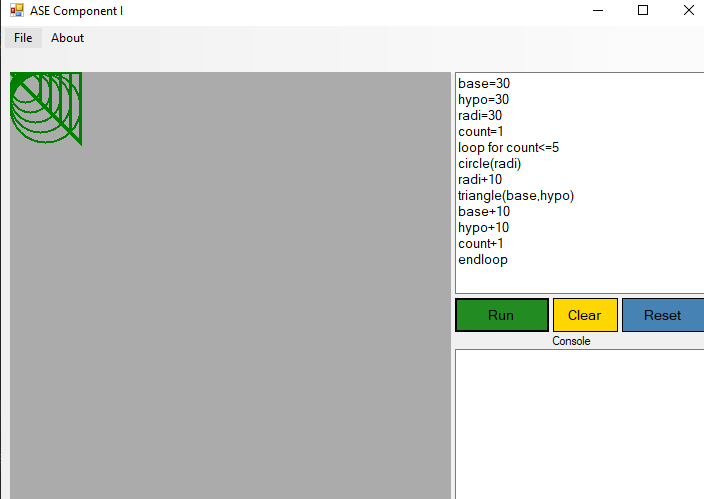
Declaring variable radius for circle which is used in parameters



Declaring variable width and height for rectangle which is used in parameters

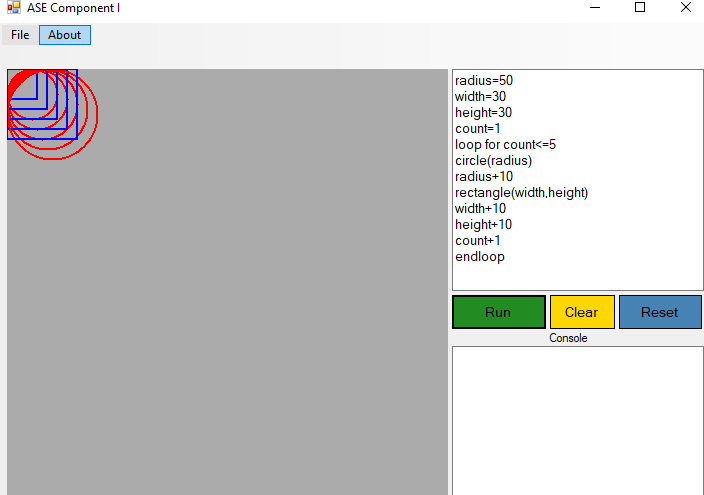


Declaring variable base and hypotenuse for triangle which is used in parameters



Example showing variable base, hypo, radi, count being used inside loops

**Loop command repeats everything between Loop on the first line and “end” on a later**

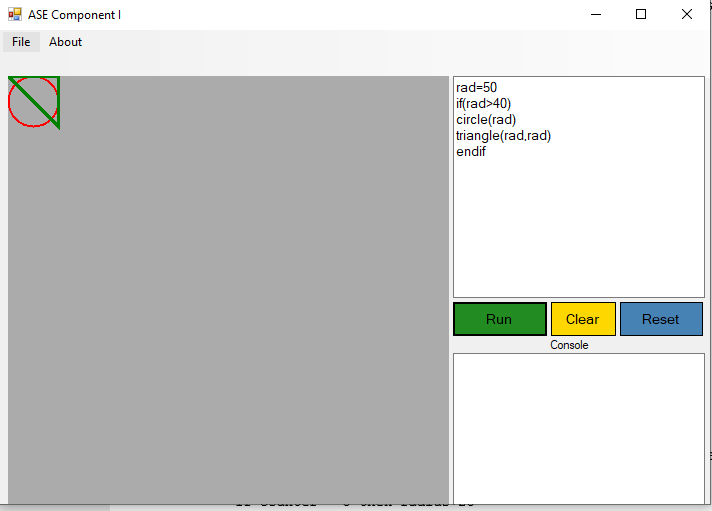


Loop command is repeating the action here 5 times drawing rectangle and circle end loop ensures that the loop has been closed



**Loop command is drawing triangle and rectangle 5 times**

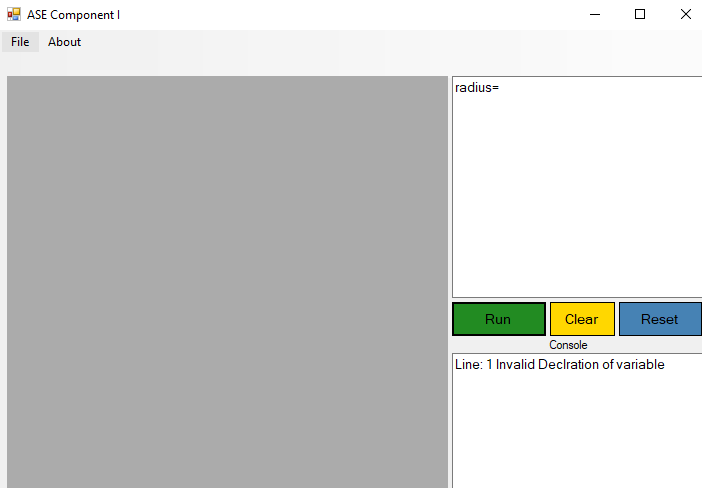
**If statement and block end with endif**



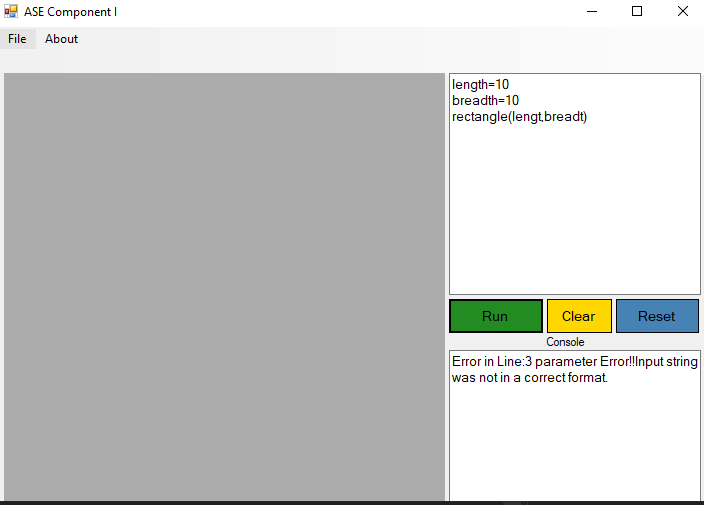
Use of if and endif opens the if statement and endif ensures that if statement is closed if checks whether radius is more than 40

**Syntax checking**

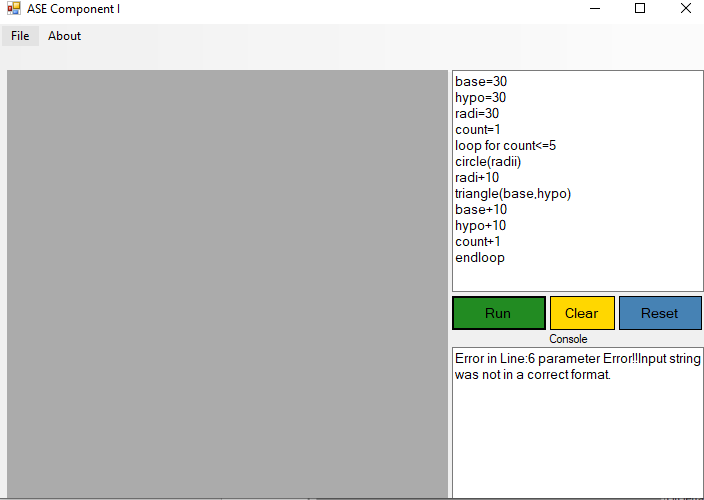
**Variable which can be used inside parameter and loop validation**



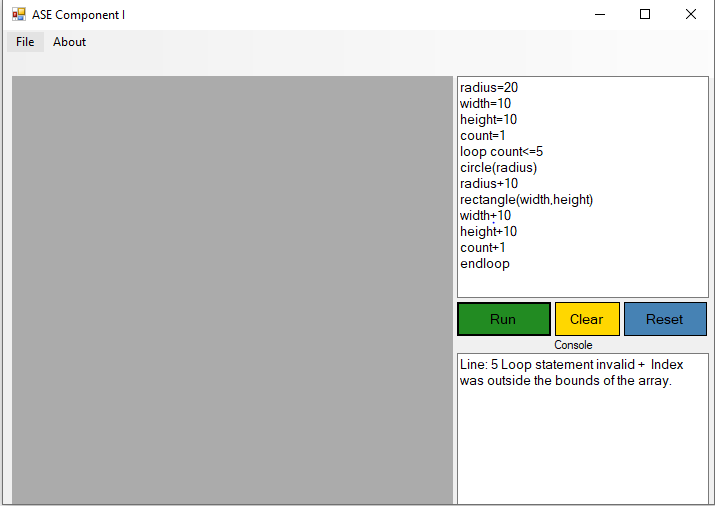
**Invalid declaration of variable**



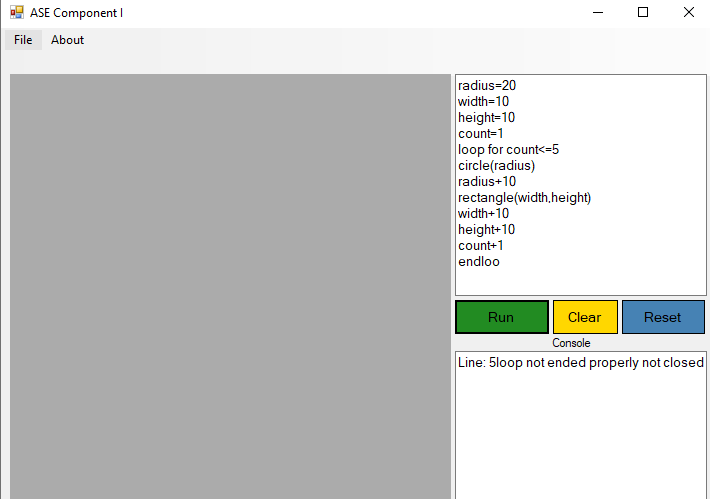
**Error shown when not declared variable is passed in a parameter**



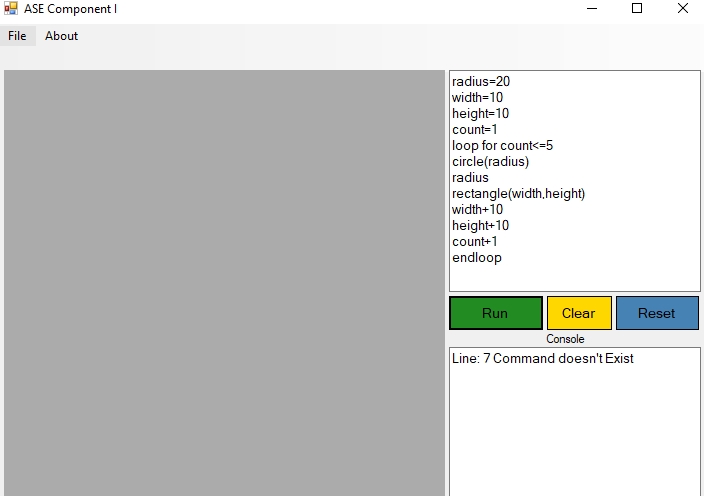
**Error shown when parameter does not found declared variable inside loop statement**

**Loop Statement Validation**

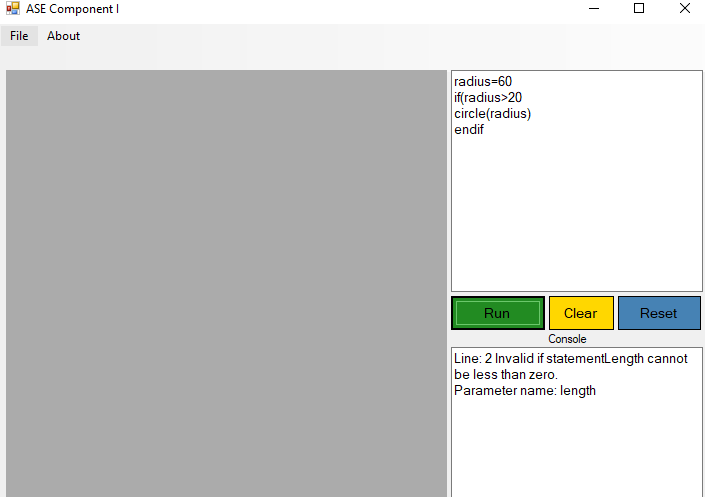
**Showing error when loop statement is invalid**



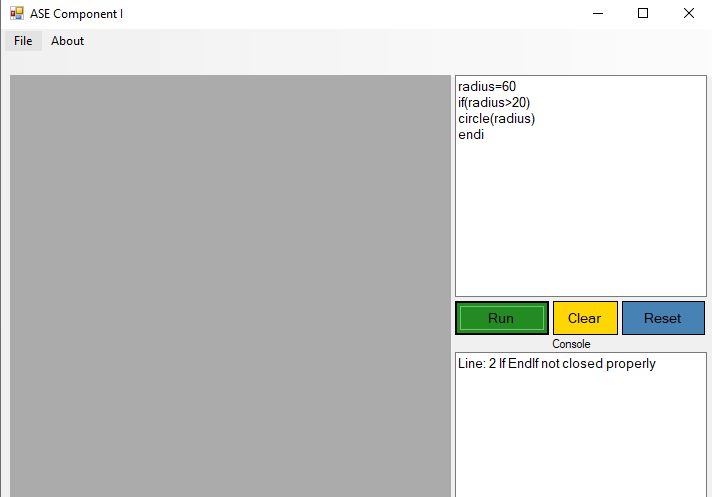
**Showing error when loop command is not closed properly**



**Error shown when increment syntax is wrong**

**If statement Validation**

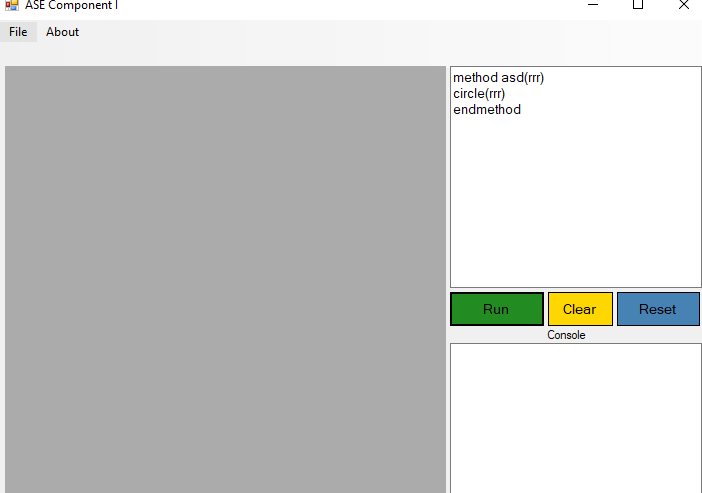
**Invalid if statement**



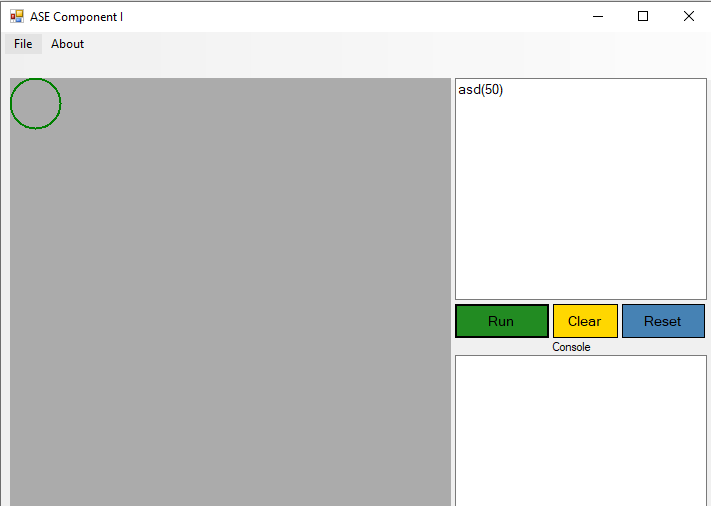
**If statement not ended properly**

**Method**

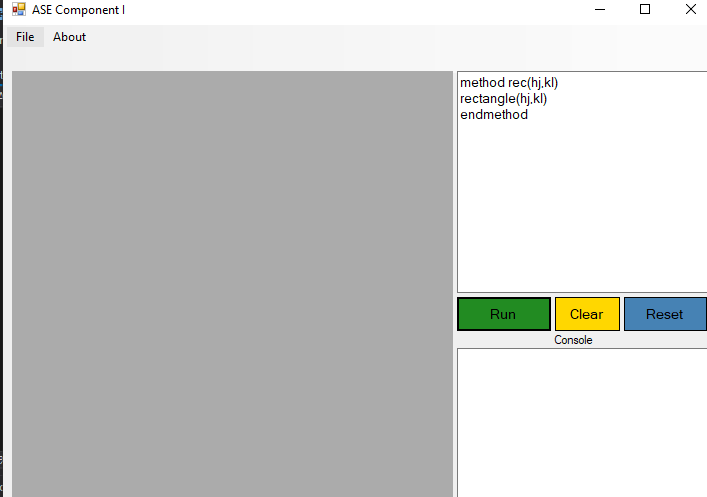
**Passing parameter**



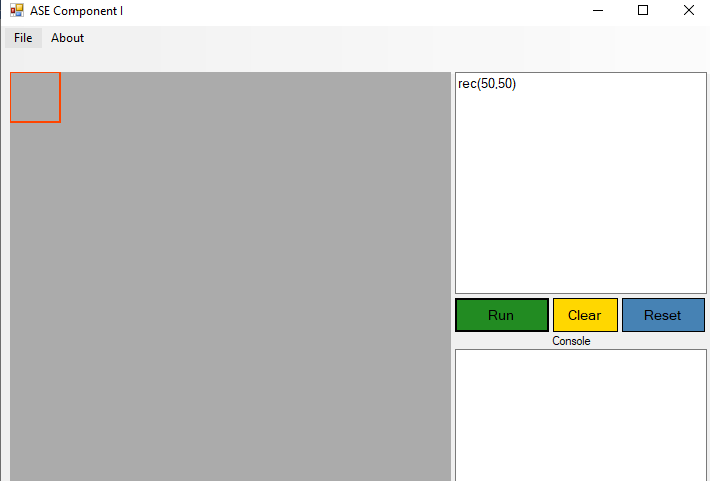
**Creating method asd giving parameter rrr and passing it in circle end method ensures that method is closed.**



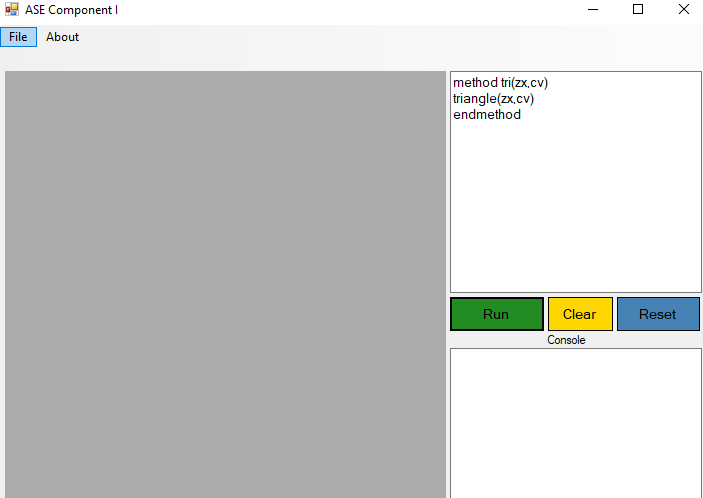
**Calling created method asd and passing 50 as parameter**



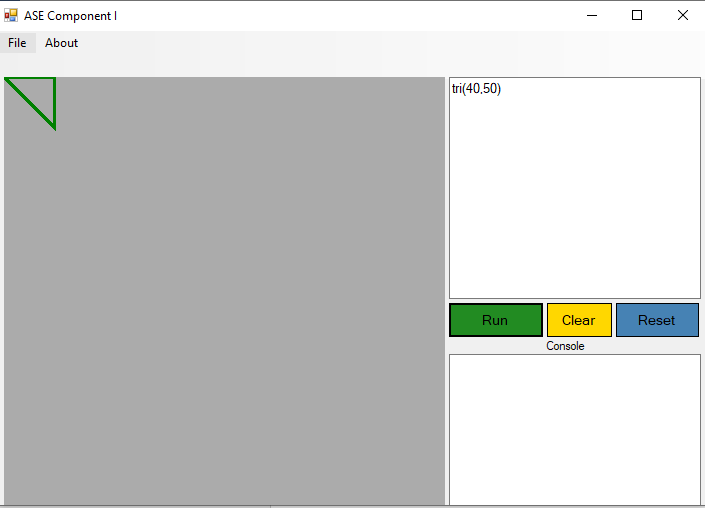
**Creating method rec giving parameter hj and kl and passing it in circle end method ensures that method is closed.**



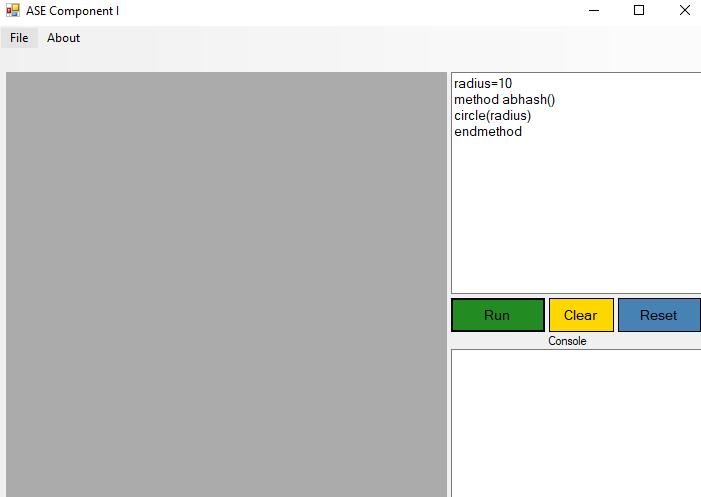
**Calling created method rec and passing 50 and 50 as parameter**



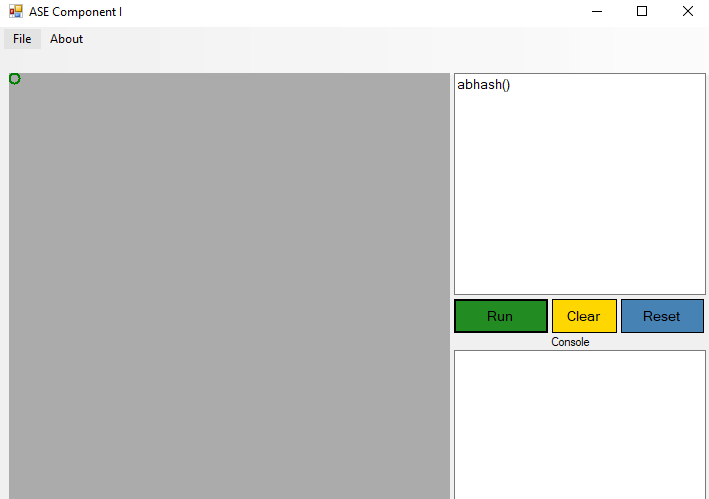
**Creating method tri giving parameter zx and cv and passing it in circle end method ensures that method is closed**



**Calling created method tri and passing 40 and 50 as parameter**

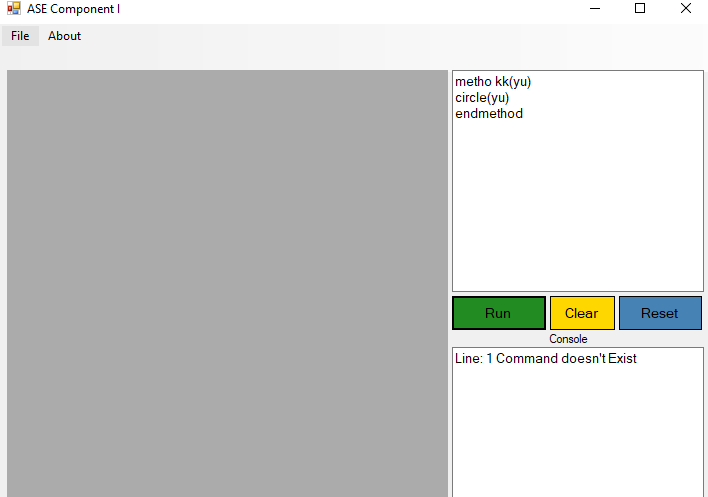


**Creating method without passing parameter**

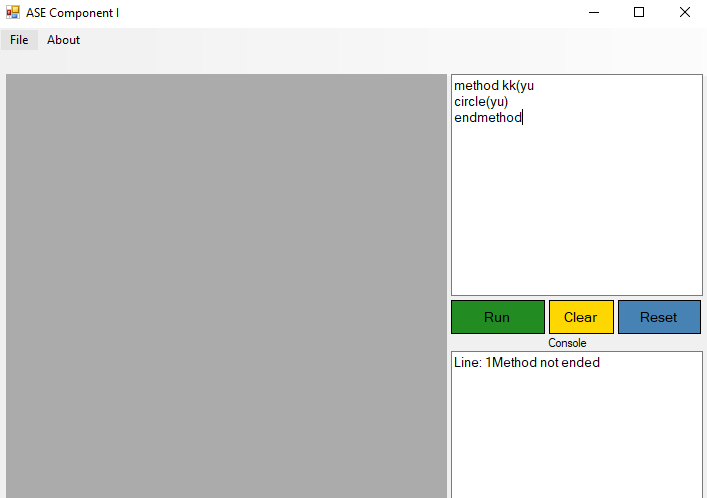


**Calling method without parameter**

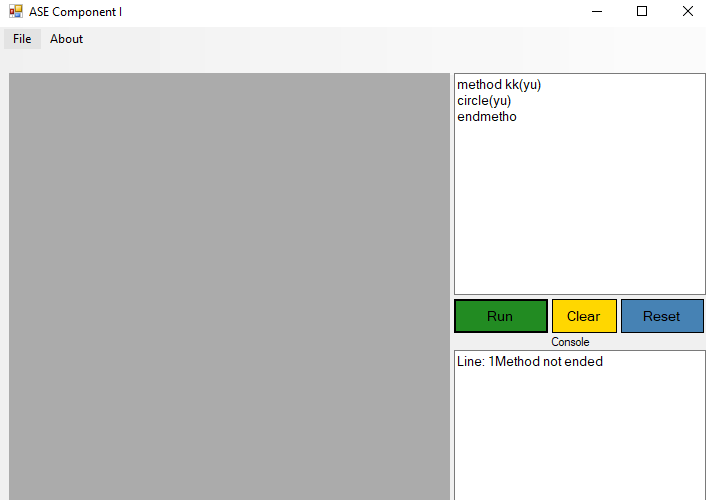
**Method Validation**



**This error is shown because metho command does not exists**

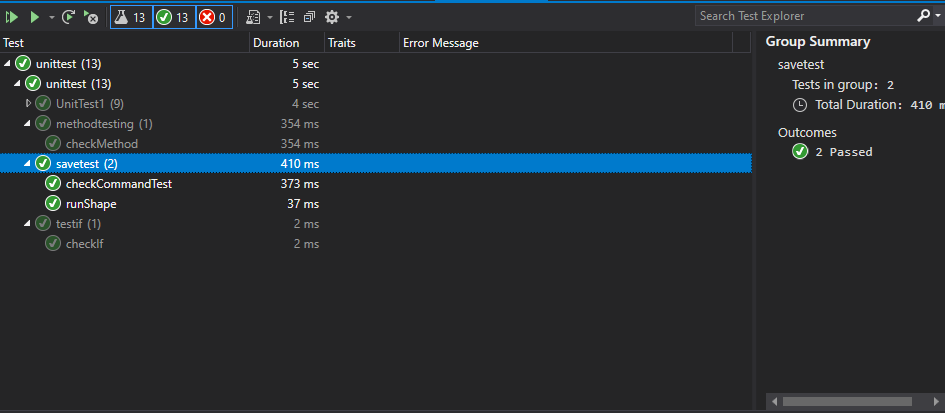


**Above error show that method is not ended properly**

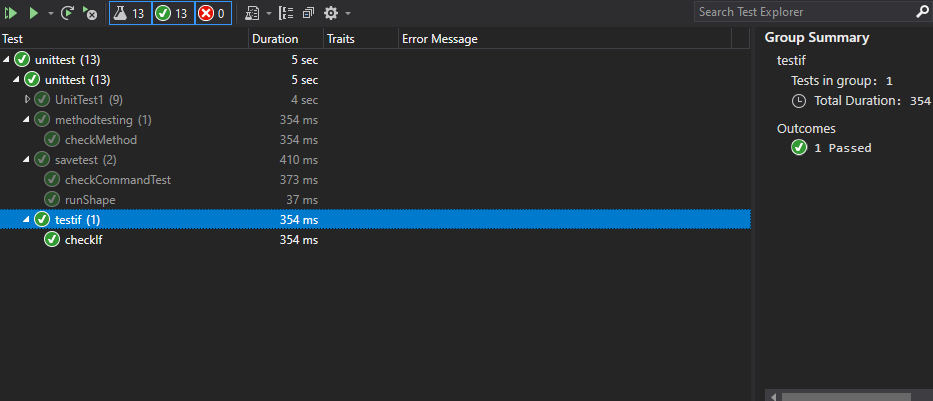


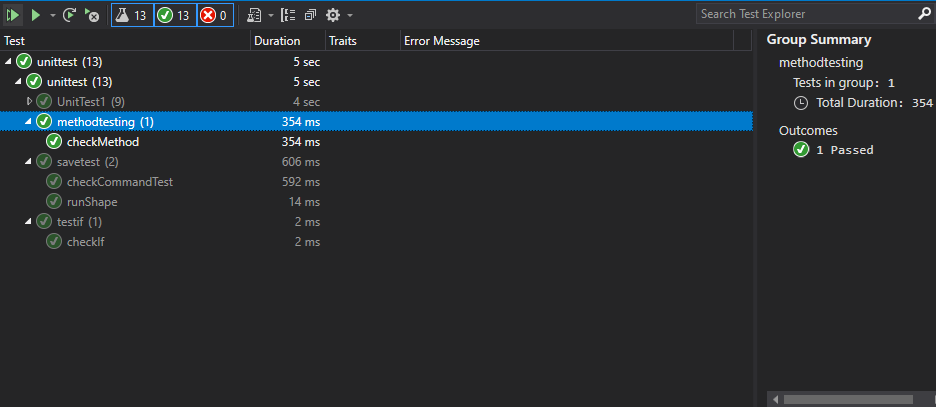
**This error is shown because end method is not closed properly**

**Unit testing**

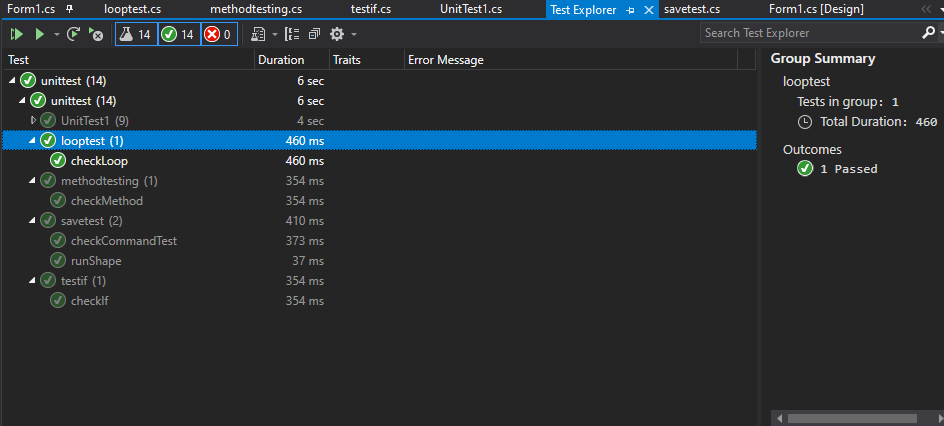


**Unit testing for check command method and run shape methods**

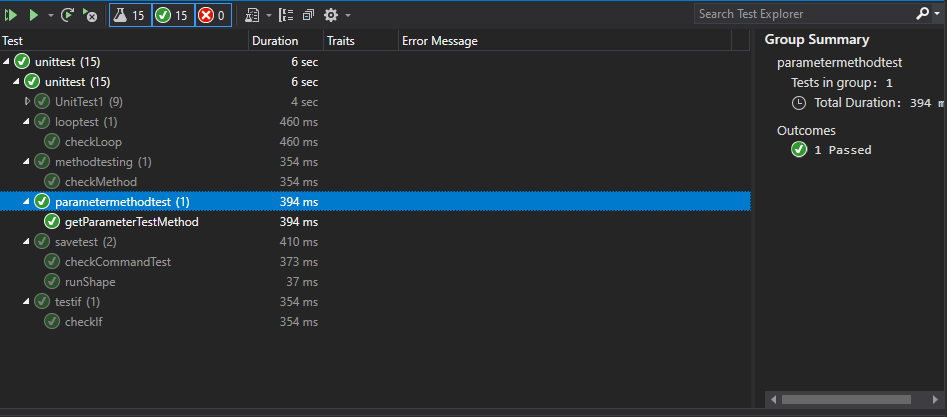
**Unit testing for check if method**



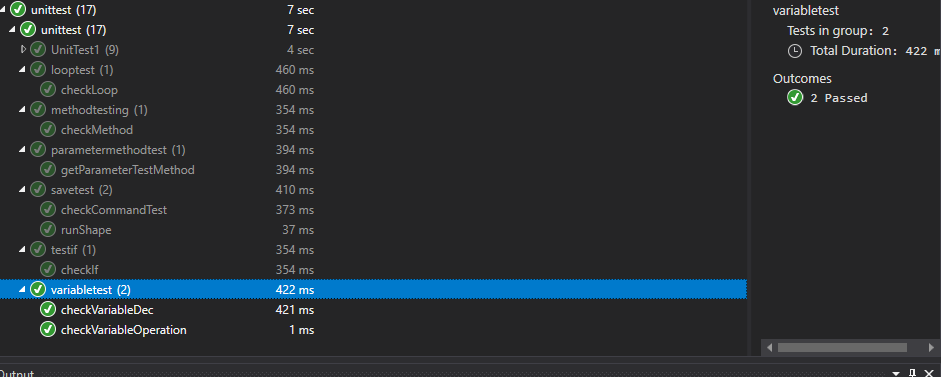
**Unit testing for check method**



**Unit testing for check loop method**



**Unit testing for get parameter test method**



**Unit testing for checkvariable dec and check variable operation method**

**Test Case**

