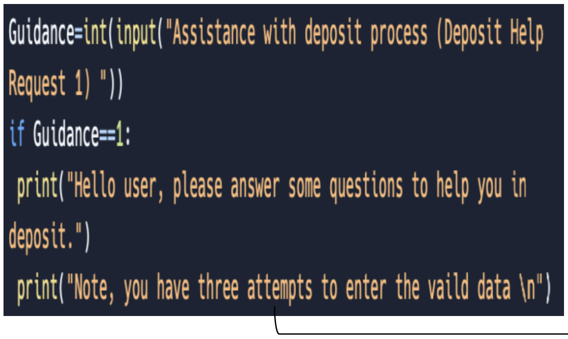
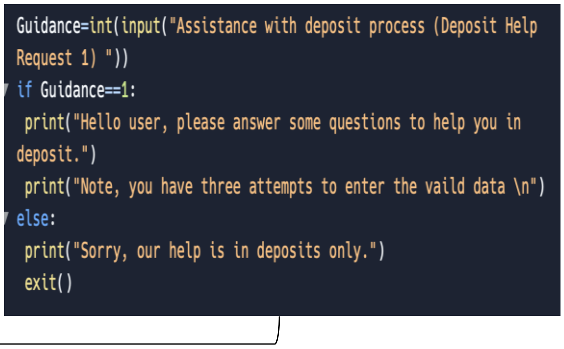
# Testing

## Testing for development

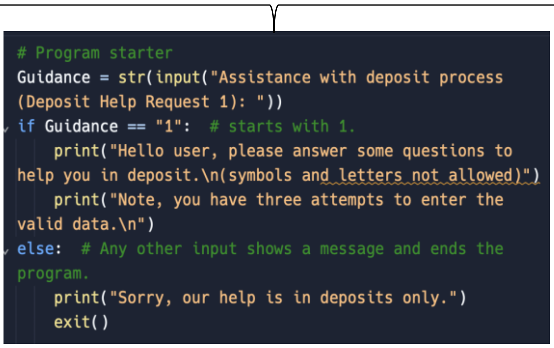


```
Guidance=int(input("Assistance with deposit process (Deposit Help
Request 1) "))
if Guidance==1:
 print("Hello user, please answer some questions to help you in
deposit.")
 print("Note, you have three attempts to enter the vaild data \n")
```



```
Guidance=int(input("Assistance with deposit process (Deposit Help
Request 1) "))
if Guidance==1:
 print("Hello user, please answer some questions to help you in
deposit.")
 print("Note, you have three attempts to enter the vaild data \n")
else:
 print("Sorry, our help is in deposits only.")
 exit()
```

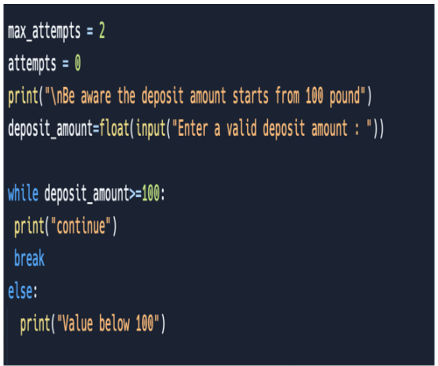
**Guidance (problem):faced when entering any value not number, which the result was error**



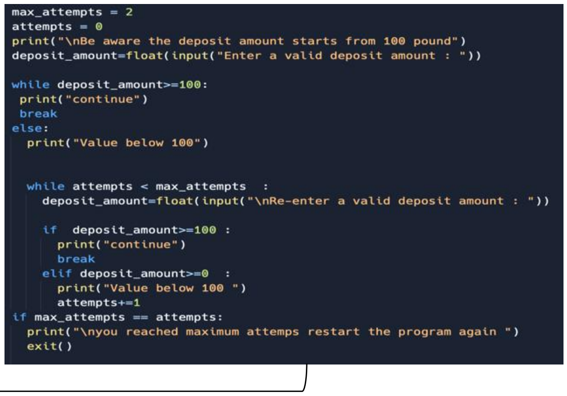
```
# Program starter
Guidance = str(input("Assistance with deposit process
(Deposit Help Request 1): "))
if Guidance == "1":  # starts with 1.
    print("Hello user, please answer some questions to
help you in deposit.\n(symbols and letters not allowed)")
    print("Note, you have three attempts to enter the
valid data.\n")
else:  # Any other input shows a message and ends the
program.
    print("Sorry, our help is in deposits only.")
    exit()
```

**I solved by change date structure from int to string to accept all options without error and with execute statements in the correct way.**

**In this development, the user was asked to enter a variable, which at first was an integer, but after devolpments, it was changed to a string to treat entering symbols or letters, and if Guidance = 1, the program will start and display a message; otherwise, the program will print a message and exit from program**

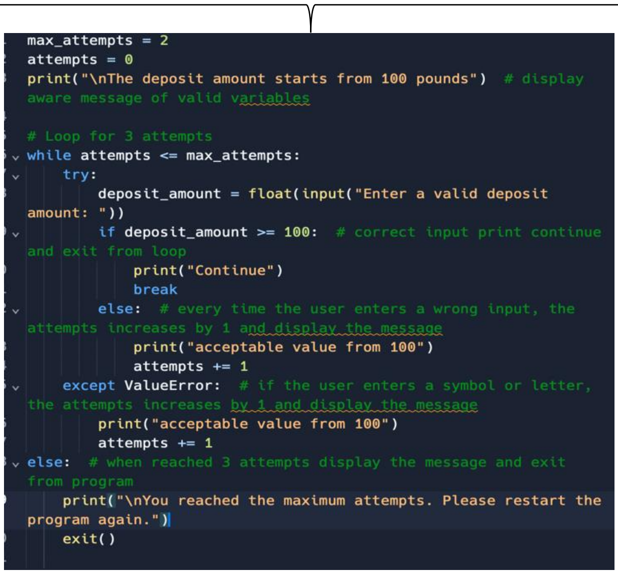


```
max_attempts = 2
attempts = 0
print("\nBe aware the deposit amount starts from 100 pound")
deposit_amount=float(input("Enter a valid deposit amount : "))

while deposit_amount>=100:
 print("continue")
 break
else:
  print("Value below 100")
```



```
max_attempts = 2
attempts = 0
print("\nBe aware the deposit amount starts from 100 pound")
deposit_amount=float(input("Enter a valid deposit amount : "))

while deposit_amount>=100:
 print("continue")
 break
else:
  print("Value below 100")


  while attempts < max_attempts   :
    deposit_amount=float(input("\nRe-enter a valid deposit amount : "))

    if  deposit_amount>=100 :
      print("continue")
      break
    elif deposit_amount>=0   :
      print("Value below 100 ")
      attempts+=1
if max_attempts == attempts:
  print("\nyou reached maximum attemps restart the program again ")
  exit()
```

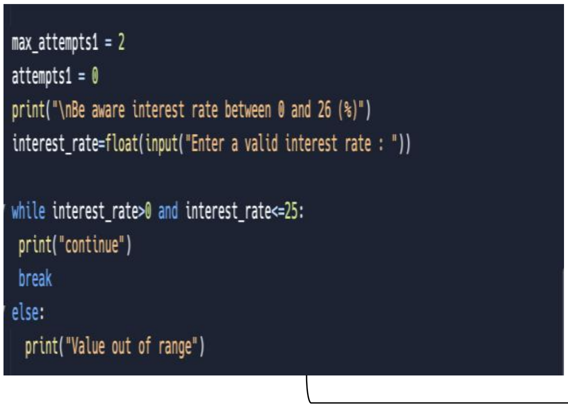
**depostit_amount(problem): faced same problem of guidance when entering any value not number, which the result was error**



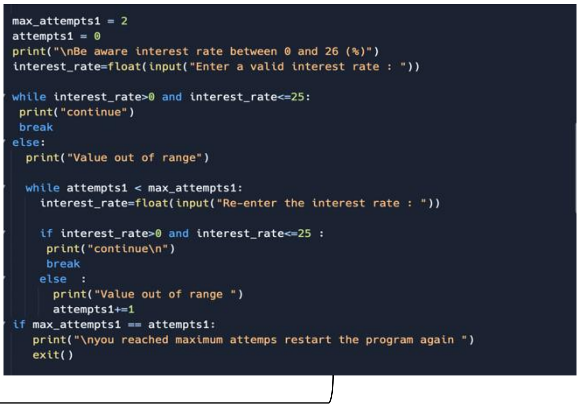
```
max_attempts = 2
attempts = 0
print("\nThe deposit amount starts from 100 pounds")  # display
aware message of valid variables

# Loop for 3 attempts
while attempts <= max_attempts:
    try:
        deposit_amount = float(input("Enter a valid deposit
amount: "))
        if deposit_amount >= 100:  # correct input print continue
and exit from loop
            print("Continue")
            break
        else:  # every time the user enters a wrong input, the
attempts increases by 1 and display the message
            print("acceptable value from 100")
            attempts += 1
    except ValueError:  # if the user enters a symbol or letter,
the attempts increases by 1 and display the message
        print("acceptable value from 100")
        attempts += 1
else:  # when reached 3 attempts display the message and exit
from program
    print("\nYou reached the maximum attempts. Please restart the
program again.")
    exit()
```

**I solved it in a different way by moving depostit_amount inside while with try statement and use except value error that prevent error to stop programme and considered invalid input as attempt.**

**User attempt variables initialized. Instruction displayed before development. Deposit amount input assigned within loop, permitting three attempts. Invalid input triggers corresponding message; reaching max attempts prompts another. Valid input breaks loop with instruction to proceed to next code segment.**

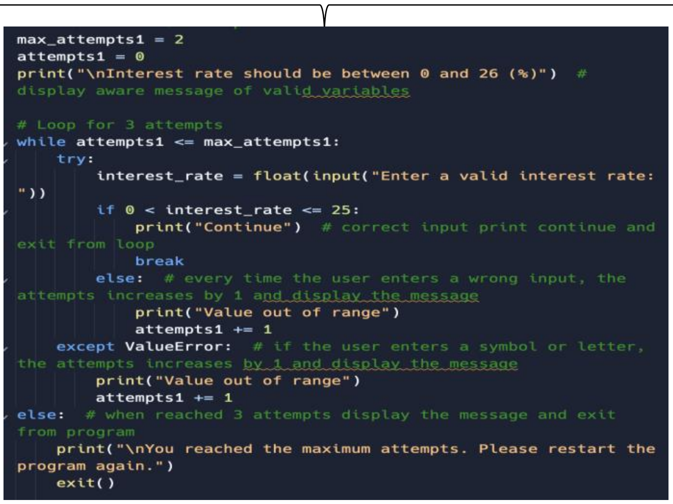


```
max_attempts1 = 2
attempts1 = 0
print("\nBe aware interest rate between 0 and 26 (%)")
interest_rate=float(input("Enter a valid interest rate : "))

while interest_rate>0 and interest_rate<=25:
 print("continue")
 break
else:
  print("Value out of range")
```



```
max_attempts1 = 2
attempts1 = 0
print("\nBe aware interest rate between 0 and 26 (%)")
interest_rate=float(input("Enter a valid interest rate : "))

while interest_rate>0 and interest_rate<=25:
 print("continue")
 break
else:
  print("Value out of range")

  while attempts1 < max_attempts1:
    interest_rate=float(input("Re-enter the interest rate : "))

    if interest_rate>0 and interest_rate<=25 :
     print("continue\n")
     break
    else  :
      print("Value out of range ")
      attempts1+=1
if max_attempts1 == attempts1:
   print("\nyou reached maximum attemps restart the program again ")
   exit()
```

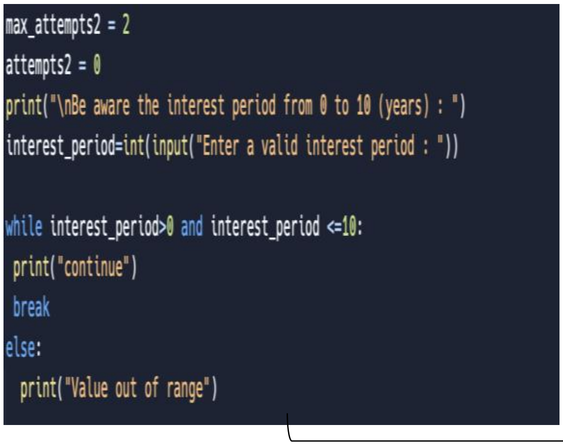
**interest_rate(problem): faced same problem of depostit_amount when entering any value not number, which the result was error**



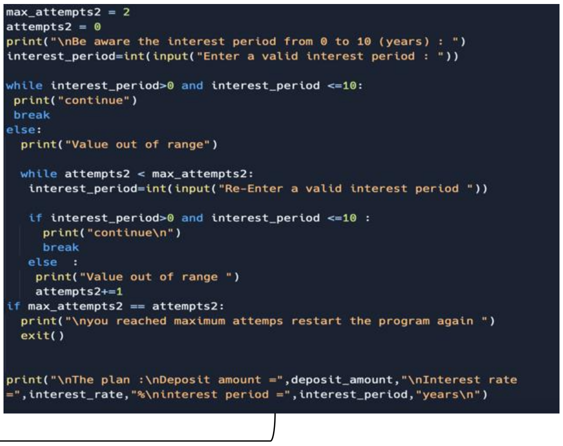
```
max_attempts1 = 2
attempts1 = 0
print("\nInterest rate should be between 0 and 26 (%)")  #
display aware message of valid variables

# Loop for 3 attempts
while attempts1 <= max_attempts1:
    try:
        interest_rate = float(input("Enter a valid interest rate:
"))
        if 0 < interest_rate <= 25:
            print("Continue")  # correct input print continue and
exit from loop
            break
        else:  # every time the user enters a wrong input, the
attempts increases by 1 and display the message
            print("Value out of range")
            attempts1 += 1
    except ValueError:  # if the user enters a symbol or letter,
the attempts increases by 1 and display the message
        print("Value out of range")
        attempts1 += 1
else:  # when reached 3 attempts display the message and exit
from program
    print("\nYou reached the maximum attempts. Please restart the
program again.")
    exit()
```

**I solved it with same way by moving interest_rate inside while with try statement and use except value error that prevent error to stop programme and considered invalid input as attempt.**

**User attempt variables initialized. Instruction displayed before development. interest rate input assigned within loop, permitting three attempts. Invalid input triggers corresponding message; reaching max attempts prompts another. Valid input breaks loop with instruction to proceed to next code segment.**

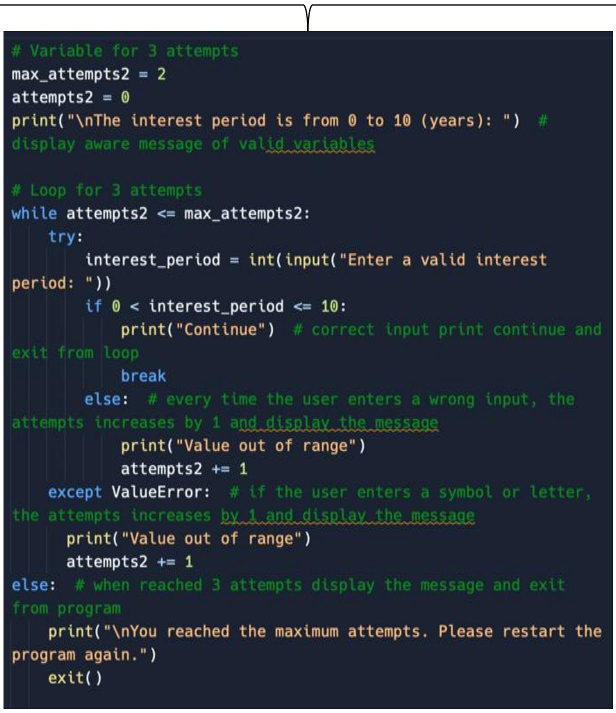


```
max_attempts2 = 2
attempts2 = 0
print("\nBe aware the interest period from 0 to 10 (years) : ")
interest_period=int(input("Enter a valid interest period : "))

while interest_period>0 and interest_period <=10:
 print("continue")
 break
else:
  print("Value out of range")
```



```
max_attempts2 = 2
attempts2 = 0
print("\nBe aware the interest period from 0 to 10 (years) : ")
interest_period=int(input("Enter a valid interest period : "))

while interest_period>0 and interest_period <=10:
 print("continue")
 break
else:
  print("Value out of range")

  while attempts2 < max_attempts2:
   interest_period=int(input("Re-Enter a valid interest period "))

   if interest_period>0 and interest_period <=10 :
     print("continue\n")
     break
   else  :
    print("Value out of range ")
    attempts2+=1
if max_attempts2 == attempts2:
  print("\nyou reached maximum attemps restart the program again ")
  exit()


print("\nThe plan :\nDeposit amount =",deposit_amount,"\nInterest rate
=",interest_rate,"%\ninterest period =",interest_period,"years\n")
```

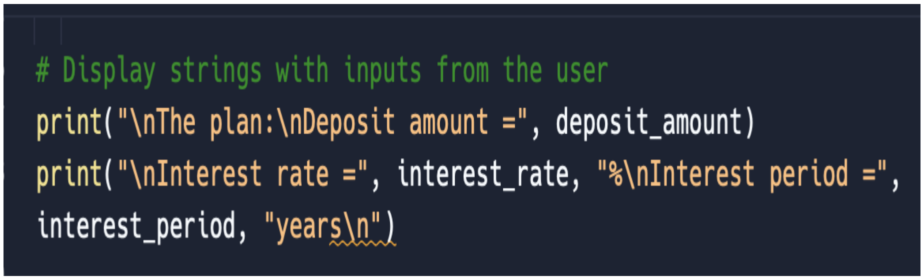
**Interest_period(problem): faced same problem of interest_rate when entering any value not number, which the result was error**



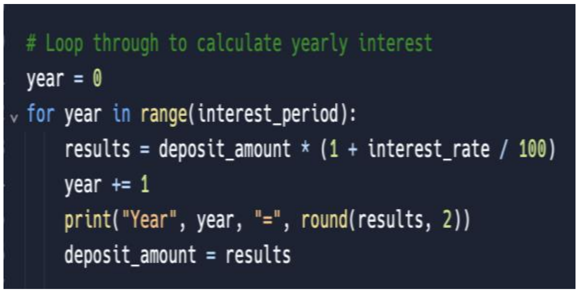
```
# Variable for 3 attempts
max_attempts2 = 2
attempts2 = 0
print("\nThe interest period is from 0 to 10 (years): ")  #
display aware message of valid variables

# Loop for 3 attempts
while attempts2 <= max_attempts2:
    try:
        interest_period = int(input("Enter a valid interest
period: "))
        if 0 < interest_period <= 10:
            print("Continue")  # correct input print continue and
exit from loop
            break
        else:  # every time the user enters a wrong input, the
attempts increases by 1 and display the message
            print("Value out of range")
            attempts2 += 1
    except ValueError:  # if the user enters a symbol or letter,
the attempts increases by 1 and display the message
      print("Value out of range")
      attempts2 += 1
else:  # when reached 3 attempts display the message and exit
from program
    print("\nYou reached the maximum attempts. Please restart the
program again.")
    exit()
```

**I solved it with same way by moving interest_period inside while with try statement and use except value error that prevent error to stop programme and considered invalid input as attempt.**

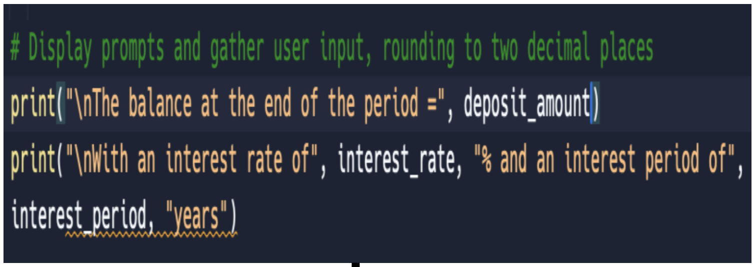
**User attempt variables initialized. Instruction displayed before development. interest period input assigned within loop, permitting three attempts. Invalid input triggers corresponding message; reaching max attempts prompts another. Valid input breaks loop with instruction to proceed to next code segment.**



```
# Display strings with inputs from the user
print("\nThe plan:\nDeposit amount =", deposit_amount)
print("\nInterest rate =", interest_rate, "%\nInterest period =",
interest_period, "years\n")
```



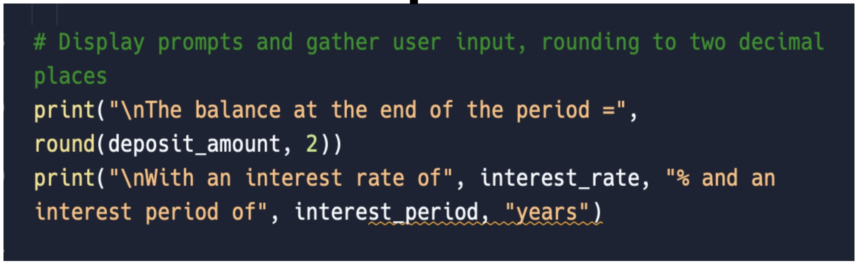
```
# Loop through to calculate yearly interest
year = 0
for year in range(interest_period):
    results = deposit_amount * (1 + interest_rate / 100)
    year += 1
    print("Year", year, "=", round(results, 2))
    deposit_amount = results
```



```
# Display prompts and gather user input, rounding to two decimal places
print("\nThe balance at the end of the period =", deposit_amount)
print("\nWith an interest rate of", interest_rate, "% and an interest period of",
interest_period, "years")
```

**Two decimal places(problem): The problem is as professional Program can’t keep float numbers to 10 decimal places**

**Two decimal places(solution): I searched for** round **function that able to specific the float number decimal places**



```
# Display prompts and gather user input, rounding to two decimal
places
print("\nThe balance at the end of the period =",
round(deposit_amount, 2))
print("\nWith an interest rate of", interest_rate, "% and an
interest period of", interest_period, "years")
```

**Initially, friendly messages displayed deposit amount, interest rate, and period. Loop assigned to calculate yearly interest with respective year and interest displayed. After enhancements, interest result rounded to two decimal places. Friendly messages about interest rate and period displayed to user.**

# Testing for Evaluation

| | Input testing | results | Test purpose |
|---|---|---|---|
| Guidance | 1 | Hello user, please answer some questions to help you in deposit.(symbols and letters not allowed)<br>Note, you have three attempts to enter the valid data. | Display the message inside if condition and start the program |
| | @ m M 2 . | Sorry, our help is in deposits only. | Display the message inside else condition and exit from the program |
| deposit_amount | >=100 | continue | Display the message inside if condition and break the loop |
| | All numbers | acceptable value from 100 | Display the message inside else condition and increase attempts by 1 |
| | @ m M 2 . | acceptable value from 100 | Display the message inside except ValueError condition and increase attempts by 1 |
| interest_rate | All decimal and integer numbers between 0 and 25 | continue | Display the message inside if condition and break the loop |
| | All decimal and integer numbers less than 0 | Value out of range | Display the message inside else condition and increase attempts by 1 |
| | @ m M . | Value out of range | Display the message inside except ValueError condition and increase attempts by 1 |
| interest_period | 0< interest_period<=10 (integers) | continue | Display the message inside if condition and break the loop |
| | interest_period>10<br>interest_period<=0 (integers) | Value out of range | Display the message inside else condition and increase attempts by 1 |
| | @ m M decimal numbers | Value out of range | Display the message inside except ValueError condition and increase attempts by 1 |

## **How the final program tested ?**

1- Run the program
2- Input testing:
-- input deposit_amount, interest_rate, and interest_period to check accept valid input
-- input incorrect values to check error handling such as specific range,letters,and symbols.

3- Test valid inputs:
-- Enter deposit amount equal or more than 100
-- Enter interest rate between 0 and 25%
-- Enter interest period between 0 and 10 inclusions (years)

4- Test invalid inputs:
-- Enter letters and symbols to any input
-- Enter deposit amount less than 100
-- Enter interest rate out of the range between 0 and 25%
-- Enter interest period out of the range between 0 and 10 inclusions (years)

5- Testing the program output
-- review the program function in yearly interest
-- display the error messages depend on conditions

6- Testing Overall Functionality
-- check if program work smoothly with user and review all conditions and messages