

Name: Mohsin Zeb

Intern ID: TN/IN02/PY/019

Email ID : mohsinzebsm@gmail.com

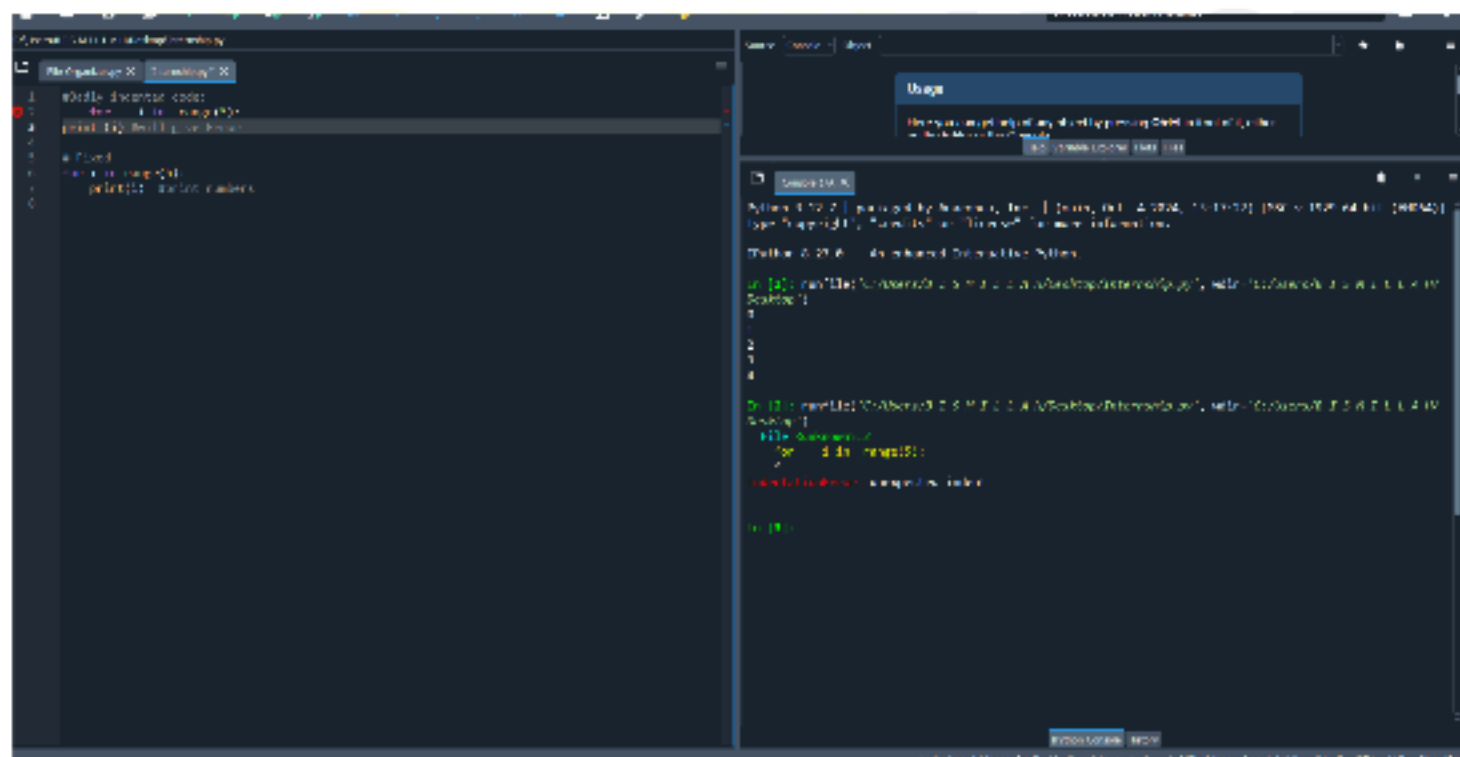
Domain: Python Development

Task no: Week #1

Question 1. Fix badly-indented code.

2. Add comments explaining each step

Code and OUTPUT:



The screenshot displays a code editor with two panes. The left pane shows the source code, and the right pane shows the output of the code execution.

Source Code (Left Pane):

```
#!/usr/bin/env python3
# This script demonstrates basic string and list operations.
# It includes comments for each step.

# Step 1: Define a string and a list
s = "Technik NEST"
l = ["Innovative", "Minds", "Nesting", "Success"]

# Step 2: Print the string and list
print(s)
print(l)

# Step 3: Concatenate the string and list
s_l = s + " " + str(l)

# Step 4: Print the concatenated string
print(s_l)
```

Output (Right Pane):

```
Technik NEST
['Innovative', 'Minds', 'Nesting', 'Success']
Technik NEST ['Innovative', 'Minds', 'Nesting', 'Success']
```

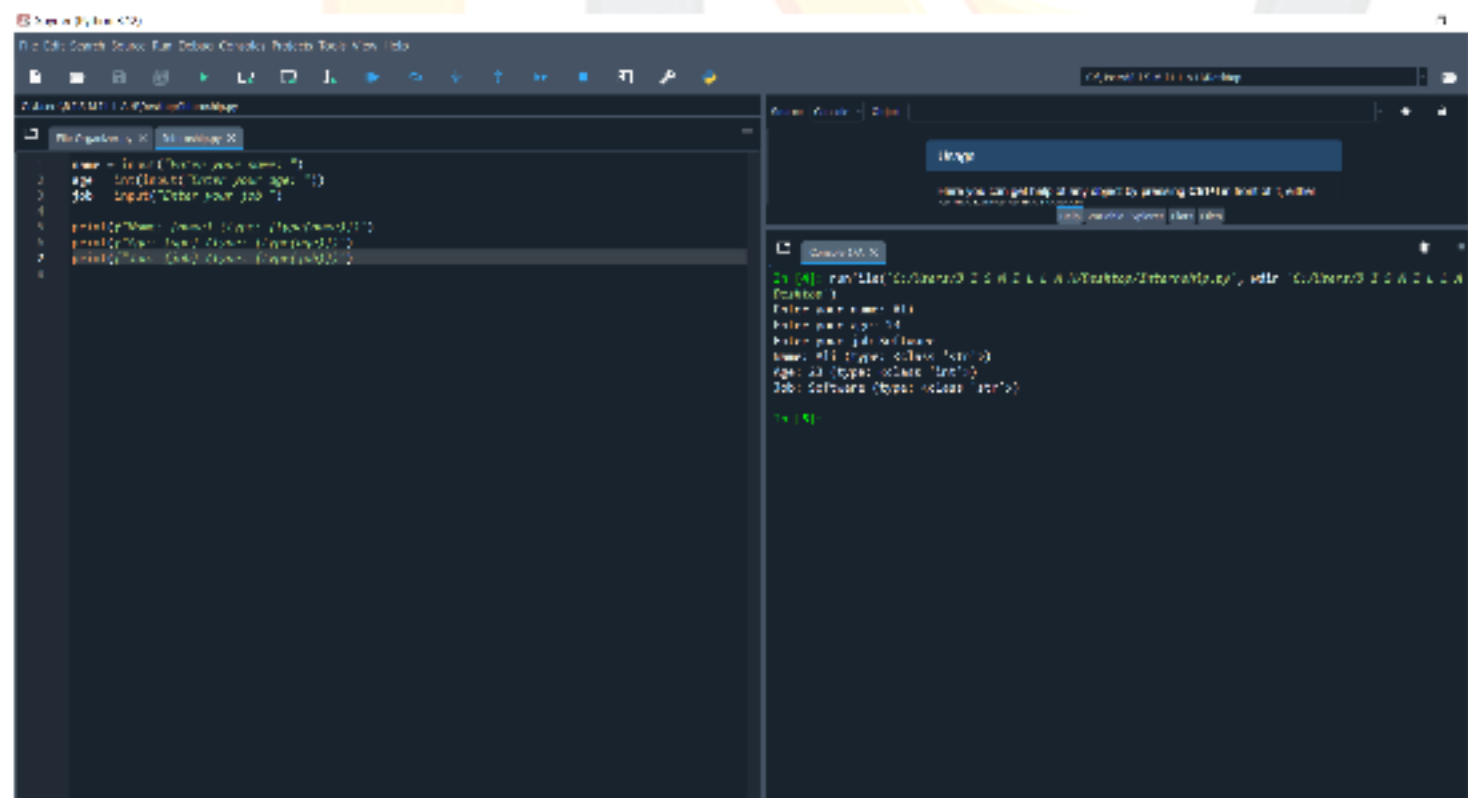
Learning:

1. Python uses indentation to define blocks of code.
2. Improper indentation causes syntax errors
3. Comments explain code to others and your future self.
4. Use # for single-line comments.

Question: 1. Collect user profile & print typed summary.

2. Swap two variables without temp var.

Code and OUTPUT:

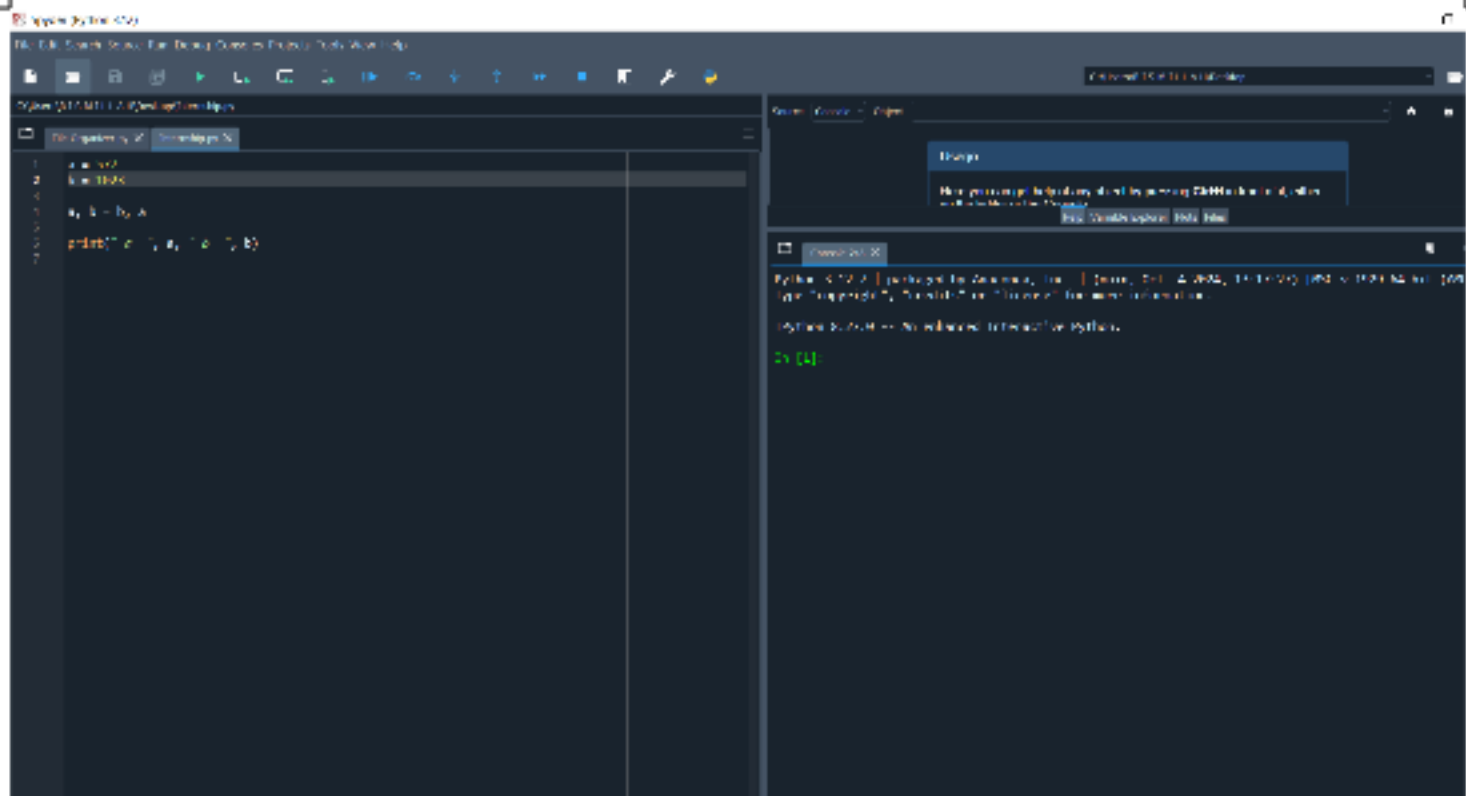


The screenshot shows a Python IDE with two windows. The left window displays the following Python code:

```
1 name = input("Enter your name: ")
2 age = input("Enter your age: ")
3 job = input("Enter your job: ")
4
5 print("Name: ", name, "Age: ", age, "Job: ", job)
6 print("Name: ", name, "Age: ", age, "Job: ", job)
7 print("Name: ", name, "Age: ", age, "Job: ", job)
```

The right window shows the output of the code, which is:

```
Enter your name: Ali
Enter your age: 34
Enter your job: Software
Name: Ali (type: <class 'str'>)
Age: 34 (type: <class 'int'>)
Job: Software (type: <class 'str'>)
```



The screenshot shows a Python IDE with a file named 'swap.py'. The code defines a function `swap(a, b)` that swaps the values of `a` and `b` using tuple unpacking. The function is called with `swap(10, 20)`, and the result is printed. The output in the console shows the values 20 and 10.

```
1 a = 10
2 b = 20
3
4 a, b = b, a
5
6 print("a = ", a, "b = ", b)
```

Output:

```
20 10
```

Learning:

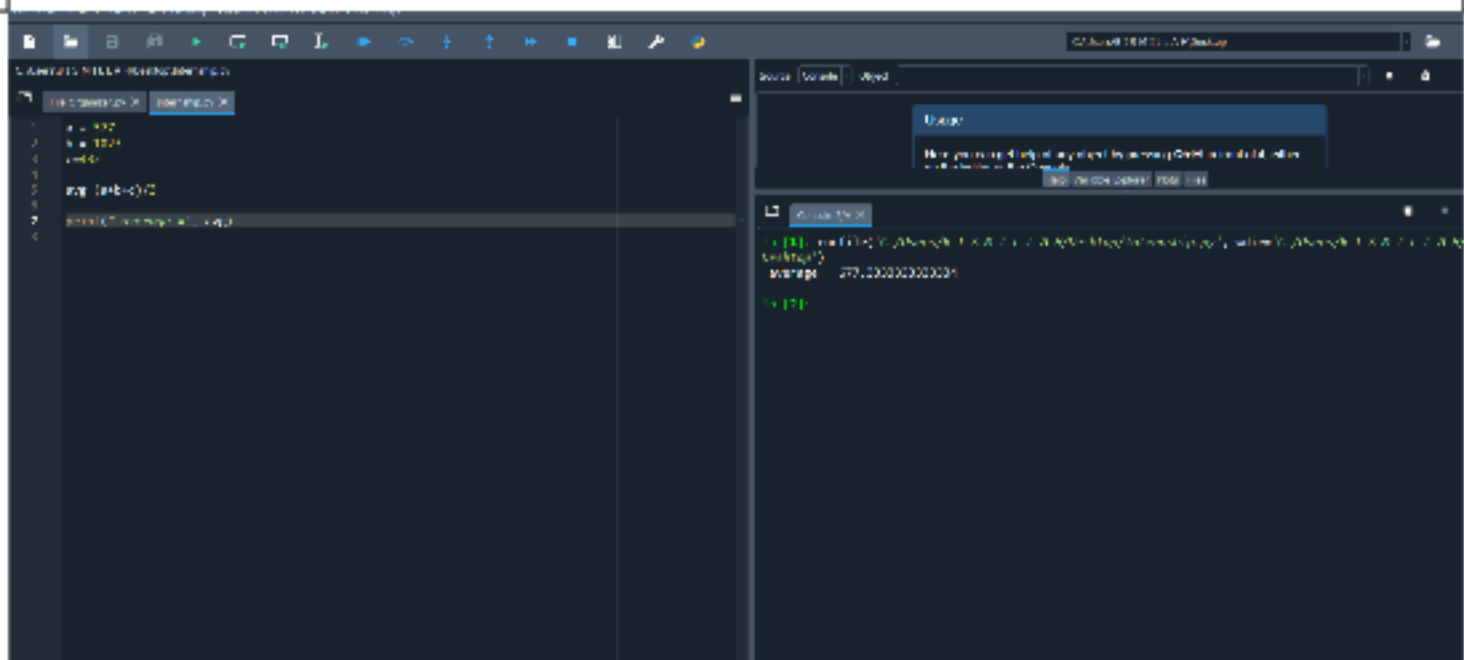
1. Variables store different types of data (int, float, str).
2. Use `input()` to get user input.
3. Swapping doesn't need a temp variable.
4. This method is clean and efficient.

TECHNIK NEST

Question: 1. Read three numbers; output avg.

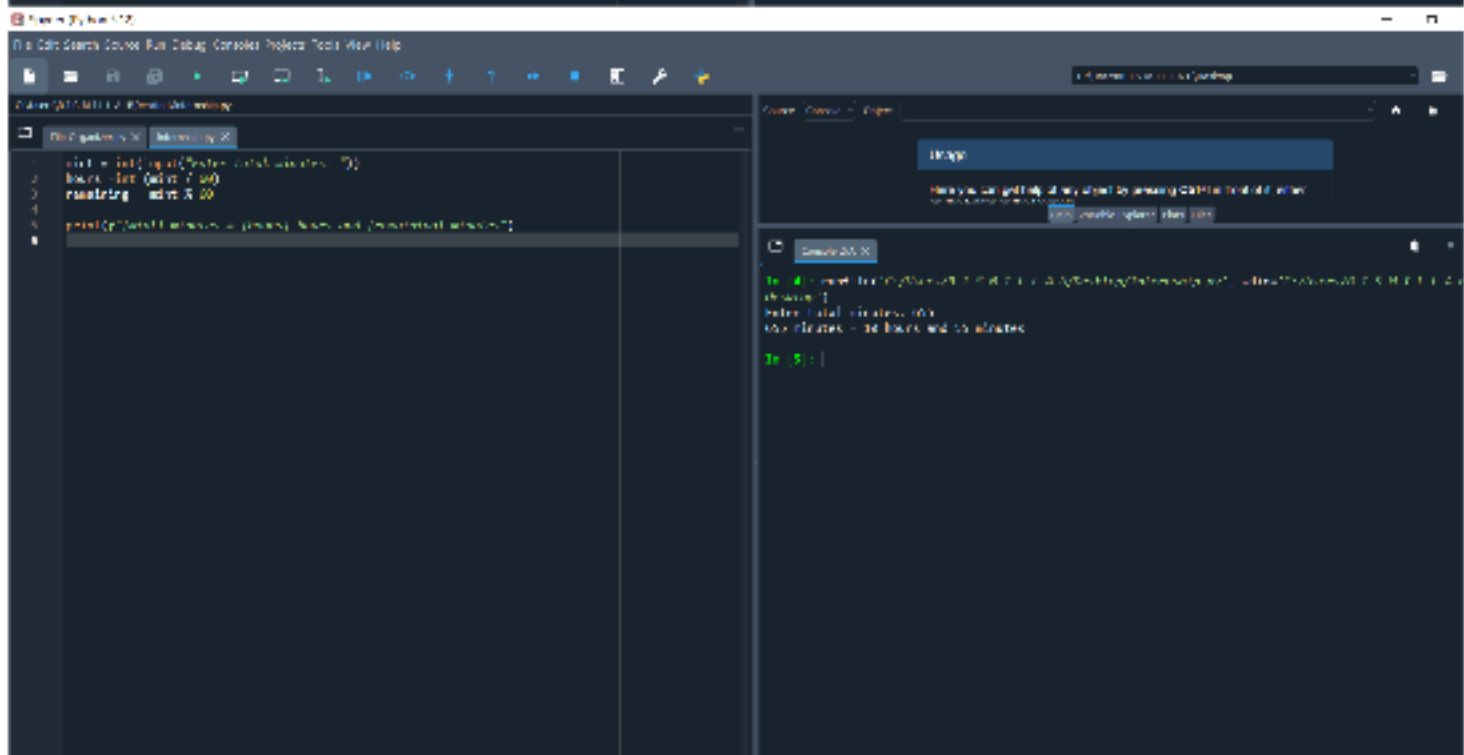
2. Convert minutes to hours + minutes.

Code and OUTPUT:



```
1 // Area of Circle Program
2 #include <iostream>
3 using namespace std;
4
5 float area(float r)
6 {
7     return 3.14 * r * r;
8 }
9
10 int main()
11 {
12     float r = 5.0;
13     float area = area(r);
14     cout << "Area of Circle is: " << area << endl;
15     return 0;
16 }
```

Output: Area of Circle is: 77.00000000000001



```
1 // Area of Circle Program
2 #include <iostream>
3 using namespace std;
4
5 #define AREA(r) 3.14 * r * r
6
7 int main()
8 {
9     float r = 5.0;
10     float area = AREA(r);
11     cout << "Area of Circle is: " << area << endl;
12     return 0;
13 }
```

Output: Area of Circle is: 77.00000000000001

TECHNIK NEST

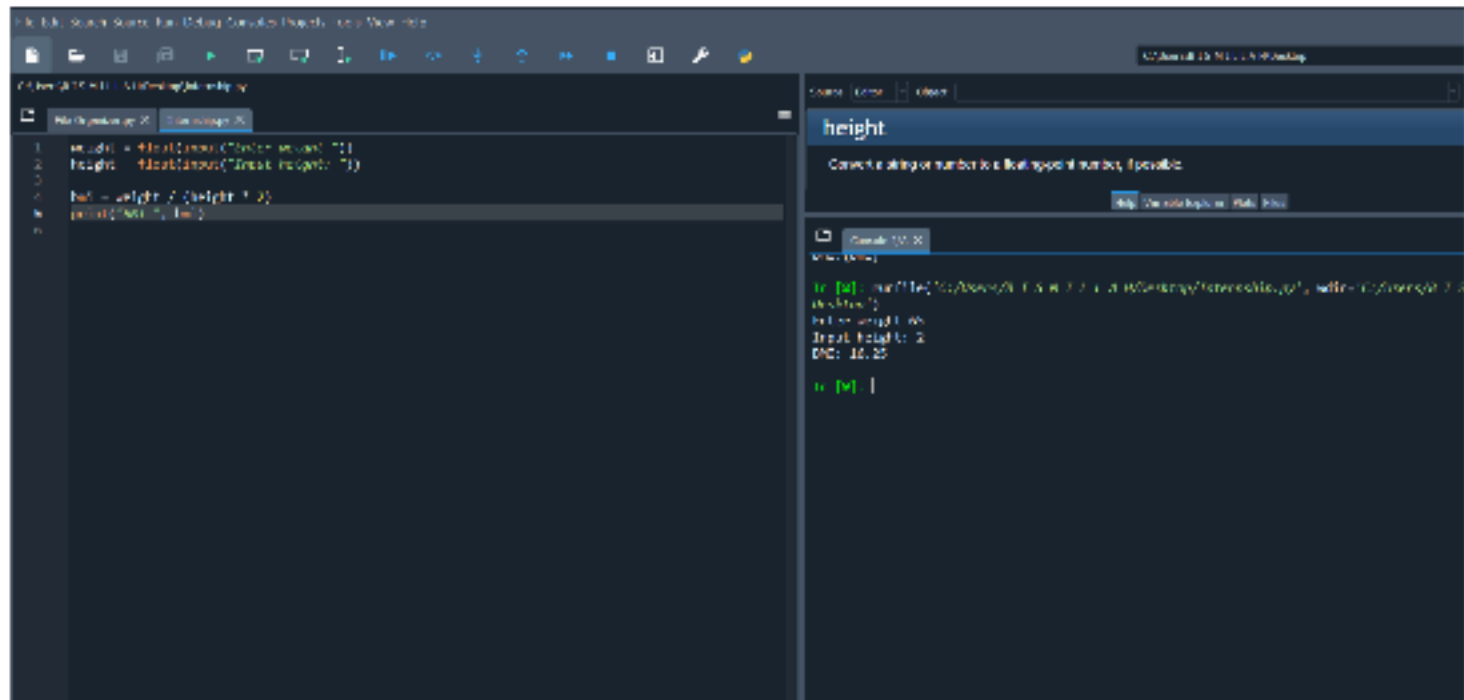
Learning:

1. Always cast numeric input using float() or int().
2. Perform arithmetic operations using operators.
3. Useful for time-based calculations.

Question: 1. BMI calc from user input.

2. Simple interest calc

Code and OUTPUT:



The screenshot shows a code editor with a file named 'BMI.js'. The code is as follows:

```
1 weight = parseFloat(prompt('weight: '));
2 height = parseFloat(prompt('height: '));
3
4 bmi = weight / (height * 2);
5 print(bmi);
```

The output console shows the following:

```
height
Convert the string of number to a float type if number, if possible
[0] 100
[1] 1.75
[2] 33.33333333333333
[3] 100
[4] 1.75
[5] 33.33333333333333
[6] 100
[7] 1.75
[8] 33.33333333333333
[9] 100
[10] 1.75
[11] 33.33333333333333
[12] 100
[13] 1.75
[14] 33.33333333333333
[15] 100
[16] 1.75
[17] 33.33333333333333
[18] 100
[19] 1.75
[20] 33.33333333333333
[21] 100
[22] 1.75
[23] 33.33333333333333
[24] 100
[25] 1.75
[26] 33.33333333333333
[27] 100
[28] 1.75
[29] 33.33333333333333
[30] 100
[31] 1.75
[32] 33.33333333333333
[33] 100
[34] 1.75
[35] 33.33333333333333
[36] 100
[37] 1.75
[38] 33.33333333333333
[39] 100
[40] 1.75
[41] 33.33333333333333
[42] 100
[43] 1.75
[44] 33.33333333333333
[45] 100
[46] 1.75
[47] 33.33333333333333
[48] 100
[49] 1.75
[50] 33.33333333333333
[51] 100
[52] 1.75
[53] 33.33333333333333
[54] 100
[55] 1.75
[56] 33.33333333333333
[57] 100
[58] 1.75
[59] 33.33333333333333
[60] 100
[61] 1.75
[62] 33.33333333333333
[63] 100
[64] 1.75
[65] 33.33333333333333
[66] 100
[67] 1.75
[68] 33.33333333333333
[69] 100
[70] 1.75
[71] 33.33333333333333
[72] 100
[73] 1.75
[74] 33.33333333333333
[75] 100
[76] 1.75
[77] 33.33333333333333
[78] 100
[79] 1.75
[80] 33.33333333333333
[81] 100
[82] 1.75
[83] 33.33333333333333
[84] 100
[85] 1.75
[86] 33.33333333333333
[87] 100
[88] 1.75
[89] 33.33333333333333
[90] 100
[91] 1.75
[92] 33.33333333333333
[93] 100
[94] 1.75
[95] 33.33333333333333
[96] 100
[97] 1.75
[98] 33.33333333333333
[99] 100
[100] 1.75
[101] 33.33333333333333
[102] 100
[103] 1.75
[104] 33.33333333333333
[105] 100
[106] 1.75
[107] 33.33333333333333
[108] 100
[109] 1.75
[110] 33.33333333333333
[111] 100
[112] 1.75
[113] 33.33333333333333
[114] 100
[115] 1.75
[116] 33.33333333333333
[117] 100
[118] 1.75
[119] 33.33333333333333
[120] 100
[121] 1.75
[122] 33.33333333333333
[123] 100
[124] 1.75
[125] 33.33333333333333
[126] 100
[127] 1.75
[128] 33.33333333333333
[129] 100
[130] 1.75
[131] 33.33333333333333
[132] 100
[133] 1.75
[134] 33.33333333333333
[135] 100
[136] 1.75
[137] 33.33333333333333
[138] 100
[139] 1.75
[140] 33.33333333333333
[141] 100
[142] 1.75
[143] 33.33333333333333
[144] 100
[145] 1.75
[146] 33.33333333333333
[147] 100
[148] 1.75
[149] 33.33333333333333
[150] 100
[151] 1.75
[152] 33.33333333333333
[153] 100
[154] 1.75
[155] 33.33333333333333
[156] 100
[157] 1.75
[158] 33.33333333333333
[159] 100
[160] 1.75
[161] 33.33333333333333
[162] 100
[163] 1.75
[164] 33.33333333333333
[165] 100
[166] 1.75
[167] 33.33333333333333
[168] 100
[169] 1.75
[170] 33.33333333333333
[171] 100
[172] 1.75
[173] 33.33333333333333
[174] 100
[175] 1.75
[176] 33.33333333333333
[177] 100
[178] 1.75
[179] 33.33333333333333
[180] 100
[181] 1.75
[182] 33.33333333333333
[183] 100
[184] 1.75
[185] 33.33333333333333
[186] 100
[187] 1.75
[188] 33.33333333333333
[189] 100
[190] 1.75
[191] 33.33333333333333
[192] 100
[193] 1.75
[194] 33.33333333333333
[195] 100
[196] 1.75
[197] 33.33333333333333
[198] 100
[199] 1.75
[200] 33.33333333333333
[201] 100
[202] 1.75
[203] 33.33333333333333
[204] 100
[205] 1.75
[206] 33.33333333333333
[207] 100
[208] 1.75
[209] 33.33333333333333
[210] 100
[211] 1.75
[212] 33.33333333333333
[213] 100
[214] 1.75
[215] 33.33333333333333
[216] 100
[217] 1.75
[218] 33.33333333333333
[219] 100
[220] 1.75
[221] 33.33333333333333
[222] 100
[223] 1.75
[224] 33.33333333333333
[225] 100
[226] 1.75
[227] 33.33333333333333
[228] 100
[229] 1.75
[230] 33.33333333333333
[231] 100
[232] 1.75
[233] 33.33333333333333
[234] 100
[235] 1.75
[236] 33.33333333333333
[237] 100
[238] 1.75
[239] 33.33333333333333
[240] 100
[241] 1.75
[242] 33.33333333333333
[243] 100
[244] 1.75
[245] 33.33333333333333
[246] 100
[247] 1.75
[248] 33.33333333333333
[249] 100
[250] 1.75
[251] 33.33333333333333
[252] 100
[253] 1.75
[254] 33.33333333333333
[255] 100
[256] 1.75
[257] 33.33333333333333
[258] 100
[259] 1.75
[260] 33.33333333333333
[261] 100
[262] 1.75
[263] 33.33333333333333
[264] 100
[265] 1.75
[266] 33.33333333333333
[267] 100
[268] 1.75
[269] 33.33333333333333
[270] 100
[271] 1.75
[272] 33.33333333333333
[273] 100
[274] 1.75
[275] 33.33333333333333
[276] 100
[277] 1.75
[278] 33.33333333333333
[279] 100
[280] 1.75
[281] 33.33333333333333
[282] 100
[283] 1.75
[284] 33.33333333333333
[285] 100
[286] 1.75
[287] 33.33333333333333
[288] 100
[289] 1.75
[290] 33.33333333333333
[291] 100
[292] 1.75
[293] 33.33333333333333
[294] 100
[295] 1.75
[296] 33.33333333333333
[297] 100
[298] 1.75
[299] 33.33333333333333
[300] 100
[301] 1.75
[302] 33.33333333333333
[303] 100
[304] 1.75
[305] 33.33333333333333
[306] 100
[307] 1.75
[308] 33.33333333333333
[309] 100
[310] 1.75
[311] 33.33333333333333
[312] 100
[313] 1.75
[314] 33.33333333333333
[315] 100
[316] 1.75
[317] 33.33333333333333
[318] 100
[319] 1.75
[320] 33.33333333333333
[321] 100
[322] 1.75
[323] 33.33333333333333
[324] 100
[325] 1.75
[326] 33.33333333333333
[327] 100
[328] 1.75
[329] 33.33333333333333
[330] 100
[331] 1.75
[332] 33.33333333333333
[333] 100
[334] 1.75
[335] 33.33333333333333
[336] 100
[337] 1.75
[338] 33.33333333333333
[339] 100
[340] 1.75
[341] 33.33333333333333
[342] 100
[343] 1.75
[344] 33.33333333333333
[345] 100
[346] 1.75
[347] 33.33333333333333
[348] 100
[349] 1.75
[350] 33.33333333333333
[351] 100
[352] 1.75
[353] 33.33333333333333
[354] 100
[355] 1.75
[356] 33.33333333333333
[357] 100
[358] 1.75
[359] 33.33333333333333
[360] 100
[361] 1.75
[362] 33.33333333333333
[363] 100
[364] 1.75
[365] 33.33333333333333
[366] 100
[367] 1.75
[368] 33.33333333333333
[369] 100
[370] 1.75
[371] 33.33333333333333
[372] 100
[373] 1.75
[374] 33.33333333333333
[375] 100
[376] 1.75
[377] 33.33333333333333
[378] 100
[379] 1.75
[380] 33.33333333333333
[381] 100
[382] 1.75
[383] 33.33333333333333
[384] 100
[385] 1.75
[386] 33.33333333333333
[387] 100
[388] 1.75
[389] 33.33333333333333
[390] 100
[391] 1.75
[392] 33.33333333333333
[393] 100
[394] 1.75
[395] 33.33333333333333
[396] 100
[397] 1.75
[398] 33.33333333333333
[399] 100
[400] 1.75
[401] 33.33333333333333
[402] 100
[403] 1.75
[404] 33.33333333333333
[405] 100
[406] 1.75
[407] 33.33333333333333
[408] 100
[409] 1.75
[410] 33.33333333333333
[411] 100
[412] 1.75
[413] 33.33333333333333
[414] 100
[415] 1.75
[416] 33.33333333333333
[417] 100
[418] 1.75
[419] 33.33333333333333
[420] 100
[421] 1.75
[422] 33.33333333333333
[423] 100
[424] 1.75
[425] 33.33333333333333
[426] 100
[427] 1.75
[428] 33.33333333333333
[429] 100
[430] 1.75
[431] 33.33333333333333
[432] 100
[433] 1.75
[434] 33.33333333333333
[435] 100
[436] 1.75
[437] 33.33333333333333
[438] 100
[439] 1.75
[440] 33.33333333333333
[441] 100
[442] 1.75
[443] 33.33333333333333
[444] 100
[445] 1.75
[446] 33.33333333333333
[447] 100
[448] 1.75
[449] 33.33333333333333
[450] 100
[451] 1.75
[452] 33.33333333333333
[453] 100
[454] 1.75
[455] 33.33333333333333
[456] 100
[457] 1.75
[458] 33.33333333333333
[459] 100
[460] 1.75
[461] 33.33333333333333
[462] 100
[463] 1.75
[464] 33.33333333333333
[465] 100
[466] 1.75
[467] 33.33333333333333
[468] 100
[469] 1.75
[470] 33.33333333333333
[471] 100
[472] 1.75
[473] 33.33333333333333
[474] 100
[475] 1.75
[476] 33.33333333333333
[477] 100
[478] 1.75
[479] 33.33333333333333
[480] 100
[481] 1.75
[482] 33.33333333333333
[483] 100
[484] 1.75
[485] 33.33333333333333
[486] 100
[487] 1.75
[488] 33.33333333333333
[489] 100
[490] 1.75
[491] 33.33333333333333
[492] 100
[493] 1.75
[494] 33.33333333333333
[495] 100
[496] 1.75
[497] 33.33333333333333
[498] 100
[499] 1.75
[500] 33.33333333333333
[501] 100
[502] 1.75
[503] 33.33333333333333
[504] 100
[505] 1.75
[506] 33.33333333333333
[507] 100
[508] 1.75
[509] 33.33333333333333
[510] 100
[511] 1.75
[512] 33.33333333333333
[513] 100
[514] 1.75
[515] 33.33333333333333
[516] 100
[517] 1.75
[518] 33.33333333333333
[519] 100
[520] 1.75
[521] 33.33333333333333
[522] 100
[523] 1.75
[524] 33.33333333333333
[525] 100
[526] 1.75
[527] 33.33333333333333
[528] 100
[529] 1.75
[530] 33.33333333333333
[531] 100
[532] 1.75
[533] 33.33333333333333
[534] 100
[535] 1.75
[536] 33.33333333333333
[537] 100
[538] 1.75
[539] 33.33333333333333
[540] 100
[541] 1.75
[542] 33.33333333333333
[543] 100
[544] 1.75
[545] 33.33333333333333
[546] 100
[547] 1.75
[548] 33.33333333333333
[549] 100
[550] 1.75
[551] 33.33333333333333
[552] 100
[553] 1.75
[554] 33.33333333333333
[555] 100
[556] 1.75
[557] 33.33333333333333
[558] 100
[559] 1.75
[560] 33.33333333333333
[561] 100
[562] 1.75
[563] 33.33333333333333
[564] 100
[565] 1.75
[566] 33.33333333333333
[567] 100
[568] 1.75
[569] 33.33333333333333
[570] 100
[571] 1.75
[572] 33.33333333333333
[573] 100
[574] 1.75
[575] 33.33333333333333
[576] 100
[577] 1.75
[578] 33.33333333333333
[579] 100
[580] 1.75
[581] 33.33333333333333
[582] 100
[583] 1.75
[584] 33.33333333333333
[585] 100
[586] 1.75
[587] 33.33333333333333
[588] 100
[589] 1.75
[590] 33.33333333333333
[591] 100
[592] 1.75
[593] 33.33333333333333
[594] 100
[595] 1.75
[596] 33.33333333333333
[597] 100
[598] 1.75
[599] 33.33333333333333
[600] 100
[601] 1.75
[602] 33.33333333333333
[603] 100
[604] 1.75
[605] 33.33333333333333
[606] 100
[607] 1.75
[608] 33.33333333333333
[609] 100
[610] 1.75
[611] 33.33333333333333
[612] 100
[613] 1.75
[614] 33.33333333333333
[615] 100
[616] 1.75
[617] 33.33333333333333
[618] 100
[619] 1.75
[620] 33.33333333333333
[621] 100
[622] 1.75
[623] 33.33333333333333
[624] 100
[625] 1.75
[626] 33.33333333333333
[627] 100
[628] 1.75
[629] 33.33333333333333
[630] 100
[631] 1.75
[632] 33.33333333333333
[633] 100
[634] 1.75
[635] 33.33333333333333
[636] 100
[637] 1.75
[638] 33.33333333333333
[639] 100
[640] 1.75
[641] 33.33333333333333
[642] 100
[643] 1.75
[644] 33.33333333333333
[645] 100
[646] 1.75
[647] 33.33333333333333
[648] 100
[649] 1.75
[650] 33.33333333333333
[651] 100
[652] 1.75
[653] 33.33333333333333
[654] 100
[655] 1.75
[656] 33.33333333333333
[657] 100
[658] 1.75
[659] 33.33333333333333
[660] 100
[661] 1.75
[662] 33.33333333333333
[663] 100
[664] 1.75
[665] 33.33333333333333
[666] 100
[667] 1.75
[668] 33.33333333333333
[669] 100
[670] 1.75
[671] 33.33333333333333
[672] 100
[673] 1.75
[674] 33.33333333333333
[675] 100
[676] 1.75
[677] 33.33333333333333
[678] 100
[679] 1.75
[680] 33.33333333333333
[681] 100
[682] 1.75
[683] 33.33333333333333
[684] 100
[685] 1.75
[686] 33.33333333333333
[687] 100
[688] 1.75
[689] 33.33333333333333
[690] 100
[691] 1.75
[692] 33.33333333333333
[693] 100
[694] 1.75
[695] 33.33333333333333
[696] 100
[697] 1.75
[698] 33.33333333333333
[699] 100
[700] 1.75
[701] 33.33333333333333
[702] 100
[703] 1.75
[704] 33.33333333333333
[705] 100
[706] 1.75
[707] 33.33333333333333
[708] 100
[709] 1.75
[710] 33.33333333333333
[711] 100
[712] 1.75
[713] 33.33333333333333
[714] 100
[715] 1.75
[716] 33.33333333333333
[717] 100
[718] 1.75
[719] 33.33333333333333
[720] 100
[721] 1.75
[722] 33.33333333333333
[723] 100
[724] 1.75
[725] 33.33333333333333
[726] 100
[727] 1.75
[728] 33.33333333333333
[729] 100
[730] 1.75
[731] 33.33333333333333
[732] 100
[733] 1.75
[734] 33.33333333333333
[735] 100
[736] 1.75
[737] 33.33333333333333
[738] 100
[739] 1.75
[740] 33.33333333333333
[741] 100
[742] 1.75
[743] 33.33333333333333
[744] 100
[745] 1.75
[746] 33.33333333333333
[747] 100
[748] 1.75
[749] 33.33333333333333
[750] 100
[751] 1.75
[752] 33.33333333333333
[753] 100
[754] 1.75
[755] 33.33333333333333
[756] 100
[757] 1.75
[758] 33.33333333333333
[759] 100
[760] 1.75
[761] 33.33333333333333
[762] 100
[763] 1.75
[764] 33.33333333333333
[765] 100
[766] 1.75
[767] 33.33333333333333
[768] 100
[769] 1.75
[770] 33.33333333333333
[771] 100
[772] 1.75
[773] 33.33333333333333
[774] 100
[775] 1.75
[776] 33.33333333333333
[777] 100
[778] 1.75
[779] 33.33333333333333
[780] 100
[781] 1.75
[782] 33.33333333333333
[783] 100
[784] 1.75
[785] 33.33333333333333
[786] 100
[787] 1.75
[788] 33.33333333333333
[789] 100
[790] 1.75
[791] 33.33333333333333
[792] 100
[793] 1.75
[794] 33.33333333333333
[795] 100
[796] 1.75
[797] 33.33333333333333
[798] 100
[799] 1.75
[800] 33.33333333333333
[801] 100
[802] 1.75
[803] 33.33333333333333
[804] 100
[805] 1.75
[806] 33.33333333333333
[807] 100
[808] 1.75
[809] 33.33333333333333
[810] 100
[811] 1.75
[812] 33.33333333333333
[813] 100
[814] 1.75
```

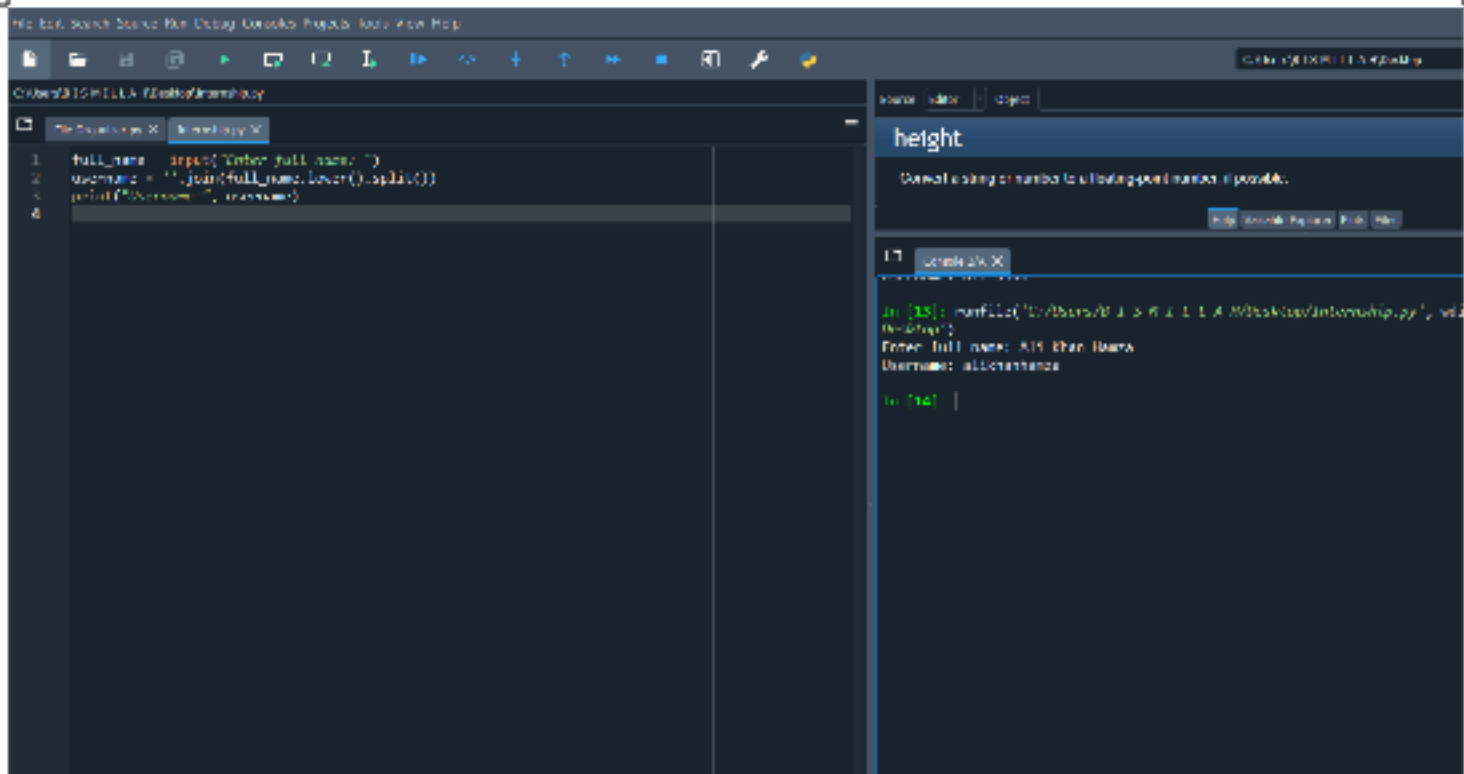
Learning:

1. BMI calculator uses height and weight to measure body reference
2. $BMI = \text{weight} / (\text{height}^2)$.
3. Order of operations matters.
4. Financial formulas can be used

Question: 1. Username builder from full name.

Code and OUTPUT:

TECHNIK NEST



The screenshot shows a Jupyter Notebook with a single cell containing the following Python code:

```
1 full_name = input("Enter full name: ")
2 username = ''.join(full_name.lower().split())
3 print("Username: ", username)
```

The output of the cell is:

```
Enter full name: All the items
Username: alltheitems
```

Learning:

1. `.lower()` converts to lowercase.
2. `.split()` breaks string into words.
3. `"".join()` joins without spaces.

Question: 1. Grade calculator.

2. Password strength classifier

Code and OUTPUT:

```
1 score = int(input("Enter your score: "))
2
3 if score > 30:
4     print("Weak password")
5 elif score > 20:
6     print("Strong password")
7 elif score > 10:
8     print("Medium password")
9 elif score > 5:
10    print("Weak password")
11 else:
12    print("Strong password")
```

height

Current testing is under the testing period and if possible

Run | Search | Python | Run | Files

Output: 20.00

```
In [14]: python3 C:\Users\A.D.S.H.E.L.A\Desktop\PasswordGenerator.py, python3 C:\Users\A.D.S.H.E.L.A\Desktop\PasswordGenerator.py
Enter your score: 40
Score: 40

In [15]: python3 C:\Users\A.D.S.H.E.L.A\Desktop\PasswordGenerator.py, python3 C:\Users\A.D.S.H.E.L.A\Desktop\PasswordGenerator.py
Enter your score: 30
Score: 30

In [16]:
```

```
1 password = input("Enter your password: ")
2
3 if len(password) > 10:
4     print("Strong password")
5 elif len(password) > 8:
6     print("Medium password")
7 else:
8     print("Weak password")
```

height

Current testing is under the testing period and if possible

Run | Search | Python | Run | Files

Output: 20.00

```
In [14]: python3 C:\Users\A.D.S.H.E.L.A\Desktop\PasswordGenerator.py, python3 C:\Users\A.D.S.H.E.L.A\Desktop\PasswordGenerator.py
Enter your password: myfirstpassword
Strong password

In [15]: python3 C:\Users\A.D.S.H.E.L.A\Desktop\PasswordGenerator.py, python3 C:\Users\A.D.S.H.E.L.A\Desktop\PasswordGenerator.py
Enter your password: myfirstpassword
Medium password

In [16]: python3 C:\Users\A.D.S.H.E.L.A\Desktop\PasswordGenerator.py, python3 C:\Users\A.D.S.H.E.L.A\Desktop\PasswordGenerator.py
Enter your password: myfirst4567
Weak password

In [17]:
```

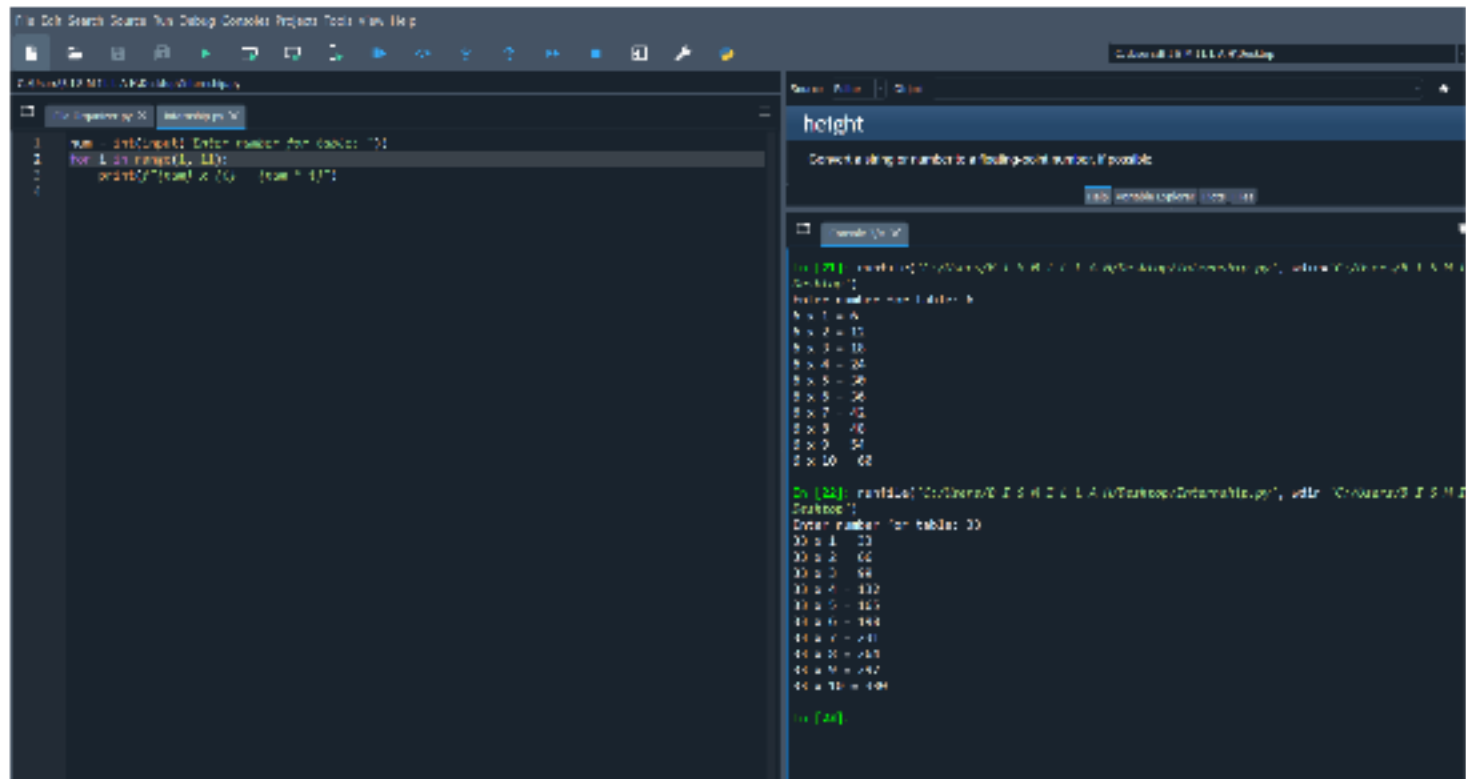
Learning:

1. if-elif-else checks multiple conditions.
2. Order matters: first true condition runs.
3. Strong passwords need length

4. Logical operators (and, or) combine conditions

Question: Multiplication Table

Code and OUTPUT:



The screenshot shows a Python IDE with a code editor on the left and a console on the right. The code in the editor is as follows:

```
1 num = int(input("Enter number for table: "))
2 for i in range(1, 11):
3     print(f"{num} x {i} = {num * i}")
```

The console output shows the program's execution. It prompts the user to enter a number, and then displays the multiplication table for the entered number (10) and a random number (20).

```
Enter number for table: 10
10 x 1 = 10
10 x 2 = 20
10 x 3 = 30
10 x 4 = 40
10 x 5 = 50
10 x 6 = 60
10 x 7 = 70
10 x 8 = 80
10 x 9 = 90
10 x 10 = 100

Enter number for table: 20
20 x 1 = 20
20 x 2 = 40
20 x 3 = 60
20 x 4 = 80
20 x 5 = 100
20 x 6 = 120
20 x 7 = 140
20 x 8 = 160
20 x 9 = 180
20 x 10 = 200
```

Learning:

1. range(start, end) generates numbers.
2. Loops are used for repetitive tasks

Question: CLI Converter:

Code and OUTPUT:

The screenshot shows a Python IDE with a file named 'unit_converter.py'. The code implements a unit converter with the following logic:

- Menu:** A list of options: 1. Convert Length (cm to m, inch to ft), 2. Convert Weight (kg to lb), 3. Convert Temperature (C to F), 4. Exit.
- Choice 1 (Length):** Prompts for 'Enter length in cm' or 'ft'. If cm, converts to m (cm/100) or ft (cm/2.54). If ft, converts to m (ft*0.3048).
- Choice 2 (Weight):** Prompts for 'Enter weight in kg'. Converts to lb (kg*2.20462).
- Choice 3 (Temperature):** Prompts for 'Enter temperature in Celsius'. Converts to Fahrenheit (C*9/5 + 32).
- Choice 4 (Exit):** Prints 'Thank you for using this program' and exits.

The output window shows the program's execution for each choice, demonstrating the conversions.

Learning:

1. Arithmetic operations can convert between units
2. Loops keep on repeating until certain conditions met
3. We can implement any calculator easily with Conditional statements