*Ethiopian Institute of architecture, building construction and city development, AAU\_ EiABC*

*Programming application short notes: Python Programming Language (part 3)*

Students should revise their study of python part one and part two by starting from interface of PyCharm covered in these part and proceed to this (python part three) part of the course. Students should refer to the appendix section of all parts of the lecture notes.

Contents

[Programming application short notes: Python Programming language (part 3) 3](#_Toc64166973)

[Python part 3 3](#_Toc64166974)

[1. Working with PyCharm 3](#_Toc64166975)

[1.1. Run 4](#_Toc64166976)

[1.2. Trace and debug 5](#_Toc64166977)

[2. User input in python command line input 7](#_Toc64166978)

[3. If, elif , else , exit and try statements in python 11](#_Toc64166979)

[4. While loop and for loop in python 20](#_Toc64166980)

[4.1. While loops 20](#_Toc64166981)

[4.2. For loops 21](#_Toc64166982)

[5. Break, continue, pass, exit, try, except 22](#_Toc64166983)

[5.1. Break 22](#_Toc64166984)

[5.2. Continue 23](#_Toc64166985)

[5.3. Pass 23](#_Toc64166986)

[6. Graphics user interface (GUI) 24](#_Toc64166987)

[6.1. Different types of Graphic user interface (GUI) for python 24](#_Toc64166988)

[6.2. Tkinter (GUI) 24](#_Toc64166989)

[6.2.1. GUI Tkinter component’s Exercises 24](#_Toc64166990)

[*6.2.2.* *GUI Class* exercises*: Grade* 27](#_Toc64166991)

[References 40](#_Toc64166992)

[Appendix 40](#_Toc64166993)

[a. Printing patterns 40](#_Toc64166994)

*List of figures*

[Figure 1: Create a new file or python file in PyCharm 3](#_Toc64166995)

[Figure 2: Project folder 4](#_Toc64166996)

[Figure 3: Running a project 4](#_Toc64166997)

[Figure 4: Break Point for debugging 6](#_Toc64166998)

[Figure 5: Step over/ Tracing and debugging 6](#_Toc64166999)

[Figure 6: Run Window for Debugging 6](#_Toc64167000)

[Figure 7: Input functions 7](#_Toc64167001)

[Figure 8: Input command 8](#_Toc64167002)

[Figure 9: User input Project browser window, code editing window and the console window 8](#_Toc64167003)

[Figure 10: Grades 29](#_Toc64167004)

[Figure 11: Exponential Growth 01 36](#_Toc64167005)

[Figure 12: Exponential growth 02 37](#_Toc64167006)

[Figure 13: Temp convert 38](#_Toc64167007)

*List of tables*

[Table 1: Trace and debug 6](#_Toc64167008)

[Table 2: User Input 7](#_Toc64167009)

[Table 3: Character, input, log and perm functions practice 8](#_Toc64167010)

[Table 4: if , elif and else in python 11](#_Toc64167011)

[Table 5: Assignment on if, elif, else and exit statement 12](#_Toc64167012)

[Table 6: While loop 21](#_Toc64167013)

[Table 7: For loop 21](#_Toc64167014)

[Table 8: GUI general 24](#_Toc64167015)

[Table 9: GUI Grids 25](#_Toc64167016)

[Table 19, Pattern 40](#_Toc64167017)

# Programming application short notes: Python Programming language (part 3)

# Python part 3

# Working with PyCharm

Refer to your python part one chapter to revise about how to open PyCharm; on creating new project and on interface of PyCharm.

PyCharm unlike IDLE, provides an opportunity to write complex and multiple line programs. Unlike IDLE you can also go back and forth through past line of codes you can make edits. Since you have already studied visual studio code project development environment, it will be easier for you to understand how PyCharm works. Text editors for programing: Kite, Vim, Atom, VS Code, sublime text and ***PyCharm***.

After you open your PyCharm environment and created a project, you can go to the project browser area right click on the project folder and select new + file or python file. File option will not give you the option of .py extension and you will have to type it in while saving. However the python file option gives you the .py extension and you do not need to type it in.

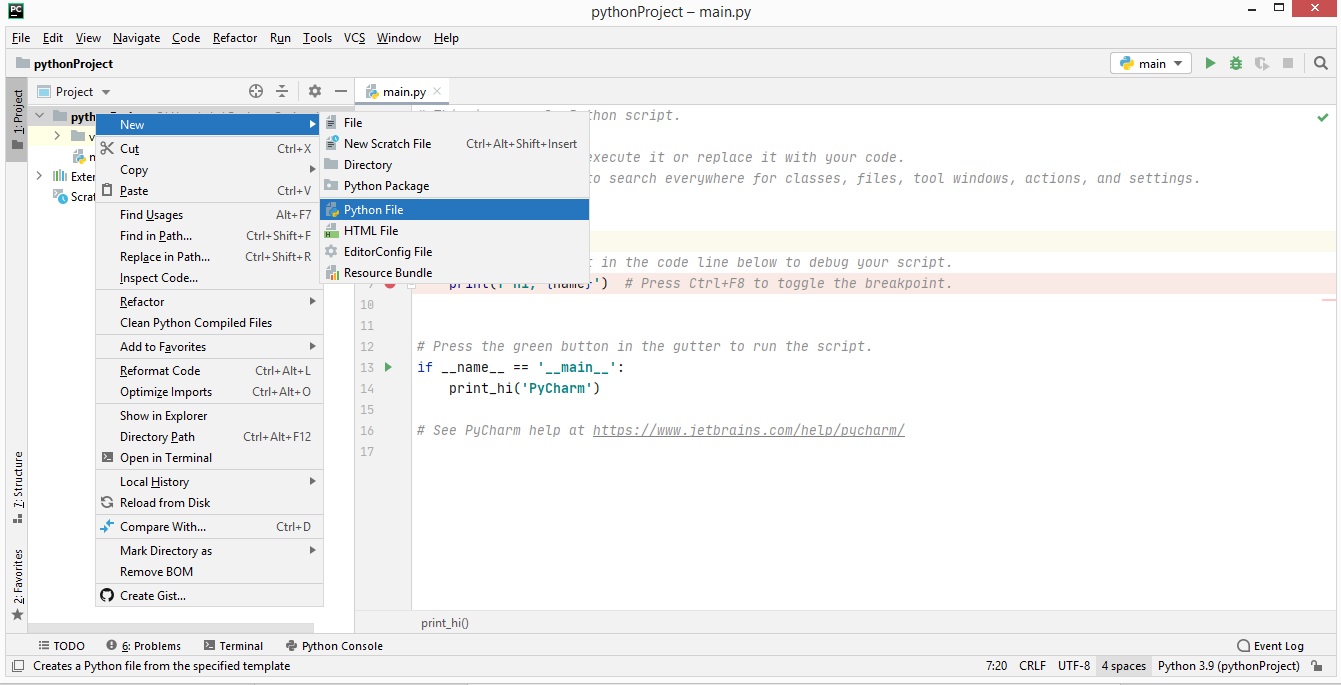


Figure : Create a new file or python file in PyCharm

On the resulting pop up window after selecting the python file option, type the name of the file as per your interest and enter. Here you do not need to type the .py in the file name as it will be provided by default.

Your project folder will be generated with different files. As you progress with your project, it will include more dependencies, modules imported, images etc.

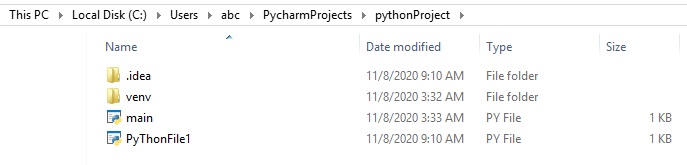


Figure : Project folder

## Run

Type in your code in the code editing area and run your program via run (menu) + run. See the following figure and compare it with the figure above.

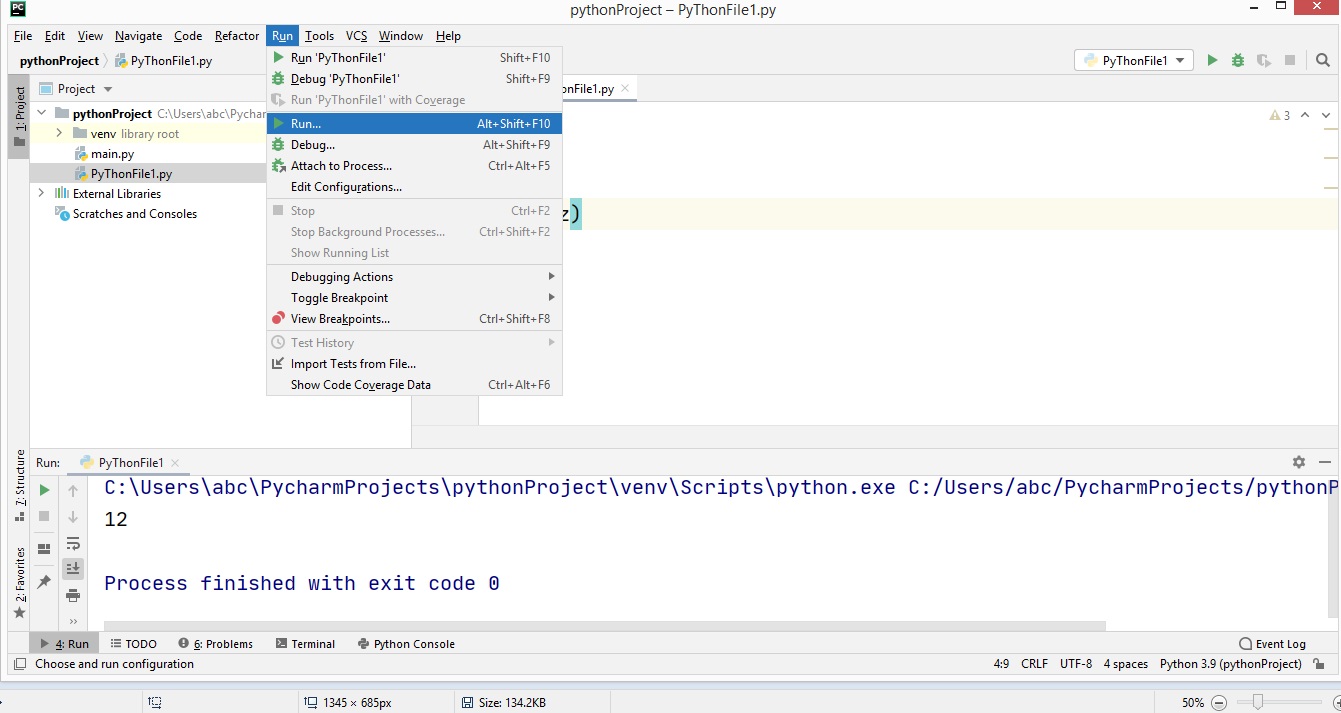


Figure : Running a project

You can also right click in your code editing area + right click + select run your python file. You can also do the same by pressing the paly button on the newly added window up on running the first time.

## Trace and debug

One of the advantages of using IDE is that you can trace and debug your cod. Tracing and debugging becomes very important when you are working with a complex multi-line codes. To trace and debug, you can go to any line on your code and click on the corresponding number. The results in a red circle dot on the selected number as in the figure below. This red dot is your break point. It serves as a starting point from where you start tracing your code for debugging. After that go to your menu and select debug. On the resulting popup window select your file and PyCharm will start debugging. To go to next line, you can press F8 or go to run (Menu) + debugging action + step over. This will allow trace and debug the next line. Check the run window on the lower part of your PyCharm window for bug information.

Table : Trace and debug

|  |  |
| --- | --- |
| Figure : Break Point for debugging | Figure : Step over/ Tracing and debugging |
| Figure : Run Window for Debugging | |

# User input in python command line input

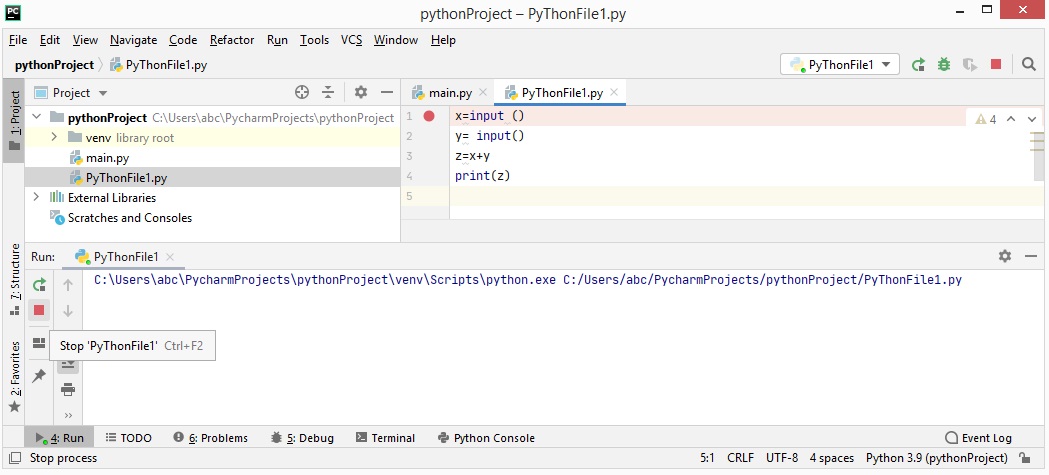


Figure : Input functions

Up until now we were running functions with variables. However we were not using an input. The input function allows you to collect client information and run it in your console window. See the following table and figures and their description.

Table : User Input

|  |  |  |
| --- | --- | --- |
| x=input() y= input() z=x+y print(z) | x=input(**"Enter Value one"**) a=int(x) y= input(**"Enter Value two"**) b=int(y) z=a+b print(z) | x=int(input(**"Enter Value one"**)) y= int(input(**"Enter Value two"**)) z=x+y print(z) |
| This input code does not yield an output in the console as there are no inputs entered. Press the stop button (red square) on the left side of the console window and enter input values. | The value you will get up on running the above code will be as strings and you need to convert the inputs to integers or floats to make use of the additive arithmetic. | Here notice that x and y are converted to integers to avoid string out put up on running the program in the console.  You can also save the line of codes you need by converting the inputs on the original variables. |
| Figure : Input command | | |

Figure : User input Project browser window, code editing window and the console window

Table : Character, input, log and perm functions practice

|  |  |
| --- | --- |
| ch=input(**'Enter a character'**) print(ch[0])  Console:  Enter a character Abebe  A | ch=input(**'Enter a character'**)[0] print(ch)  Console:  Enter a character Abebe  A |
| When you run this input function, you will always get a character. Even if the client or a user entered an entire string. This is because we are fetching the 0 index. | The index inquiry can also be typed in the first line |
| expression = eval (input(**'enter expression'**)) print(expression)  Console:  eneter expression 1000/100-10  0.0 | |
| Class exercise on math module, input, log function. | |
| On your earlier class exercise you have computed salary growth of a person in t years. By making use of the formula: Y=a(r+1)t . For this exercise, make use of the same formula and setup an input structure (as shown in class) for Y (salary in t years) in PyCharm and show your work to your instructor.  Answer:  a=float(input("Enter Basic Salary")) r=float(input("Enter Rate of salary increase")) t=float(input("Enter Year of Experience")) #y=a\*(r+1)\*\*t y=a\*pow((r+1),t) print(y)  Restructure the above formula and setup an input structure to compute for t (number of years) in PyCharm and show it to your instructor. (5%) class exercise    Answer  import math  a=float(input(**"Enter initial Salary"**)) r=float(input(**"Enter Rate of salary increase"**)) *#t=float(input("Enter Year of Experience")) #y=a\*(r+1)\*\*t #y=a\*pow((r+1),t)* y=float(input(**"Enter Current salary amount"**)) t=math.log(y/a,r+1) *#total work experience in years* print(t) | |

Assignment on math module, input and perm function

|  |  |  |
| --- | --- | --- |
| Assignment: | | |
| This exercise also has two parts and you can work the first step on your IDLE environment and the second step n PyCharm and prepare two name.py files for the exercise.  The exercise is as follows: students that are not well versed with permutations and combinations may use the descriptions provided in the boxes under each question. The responsibility of the students is to make use of their python programing language and be able to access module and function like math, log, sqrt, perm, etc. to carry out the task. Please mind all components of the descriptions provided. (! In your standard algebra means factorial) | | |
| Step 1 = what are the chances that no two boys are sitting together for a photograph if there are five girls and two boys? (once they are done working with this step students should save name.py of the file and move on to the next step of the exercise on a new python file) | | |
| Description:  Here we can apply (a) permutation and combination and (b) probability.  Total number of children are 7. Yielding a total of 7! Total Ways of sitting combinations. Let us say: total number of ways children can sit together=CST, total number of ways that two boys can sit together=BST, total number of ways no two boys can sit together=NBST; therefore; CST=BST+NBST ; probability of no two boys sitting together =PNBST  CST=7! *#7 is the total number of children = tnc = b + g*  BST=2! X 6! *#2 is the number of boys that cannot sit together = nbnt and 6 is = tnc -1*  NBST=CST-BST *# 7!-2!x6!=7x6!-2!x6!=6!x5 … the workout is not a necessary information for your task. You only need the formula that can be computed in IDLE or PyCharm.*  PNBST=NBST/CST *# 6!x5/7!=6!x5/7x6!=5/7=0.7142 … the workout is not a necessary information for your task. You only need the formula that can be computed in IDLE or PyCharm.*  The comment lines are placed for you because you will need them to work on the second step of this exercise. | | |
| Answer for step 01, | | |
| IDLE | PyCharm | |
| >>> import math  >>> CST=math.perm(7)  >>> BST=math.perm(2)\*math.perm(6)  >>> NBST=CST-BST  >>> PNBST=NBST/CST  >>> PNBST  0.7142857142857143 | import math CST=math.perm(7) BST=math.perm(2)\*math.perm(6) NBST=CST-BST PNBST=NBST/CST print(PNBST) | |
| step2 = for the scenarios in step one create a variable for number of boys (for example: b) , for number of girls (for example: g), total number of children (for example: tnc) and another variable for number of boys that cannot sit together (for example: nbnt). Model a scenario by making use of the input function to compute probability of no boys sitting together (PNBST) for any appropriate value that is entered for tnc or nbnt. (once students are done working on this file students shouldsave name.py file of the exercise) | | |
| Answer for step 02, | | |
| PyCharm | | |
| import math b=int(input(**'enter number of boys'**)) g=int(input(**'enter number of girls'**)) tnc=b+g nbnt=int(input(**'enter the number of boys that cannot sit together'**)) CST=math.perm(tnc) *#tnc is the totalnumber of children* BST=math.perm(nbnt)\*math.perm(tnc-1) *#nbnt is the number of boys that can not sit togehter* NBST=CST-BST PNBST=NBST/CST if PNBST>=0 and PNBST<=1:  print(PNBST) else:  print(**'invalid or there is no solution'**) | | |
| Or | | |
| Step 01 | | Sep 02 |
| import math CST=math.factorial(7) BST=math.factorial(2)\*math.factorial(6) NBST=CST-BST PNBST=NBST/CST print(**'Probability of No Two Boys Sit Together is ='**,PNBST) | | import math b=int(input(**'Please enter number of boys'**)) g=int(input(**'Please enter number of girls'**)) tnc=b+g nbnt=int(input(**'Please enter number of boys not to sit together'**)) CST=math.factorial(tnc) BST=math.factorial(nbnt)\*math.factorial(tnc-1) NBST=CST-BST PNBST=NBST/CST print(**'Probability of no '**,nbnt,**'Boys sit together is = '**,PNBST) |

# If, elif , else , exit and try statements in python

The enteral processing unit of our computer employs the principles of the ‘if’ key to process programs in its logical unit section. CPU (central processing unit) = CU (Control unit) + MU (memory Unit) + ALU (Arithmetic unit and logical unit). Knowing flowchart helps for programing (reading).

Mind the colon syntax in the following exercises.

Table : if , elif and else in python

|  |  |
| --- | --- |
| if | If , else |
| *# mind the indentation / commonly four.* if False:  print(**"all is well"**) if True:  print(**"all is well!"**) x=11 y= x % 2 if y==0:  print(**'Even number'**) if y==1:  print(**'Odd number'**) | *# mind the indentation / commonly four. # you can make your programming efficient by making use of the else key*  if False:  print(**"all is well"**) else:  print(**"all is well!"**) x=11 y= x % 2 if y==0:  print(**'Even number'**) else:  print(**'Odd number'**) |
| *# mind the indentation / commonly four. # you can make your programming efficient by making use of the else key # when you have if within if, the second if is considered if the second if is satisfied. This case is called nested if.*  if False:  print(**"all is well"**) else:  print(**"all is well!"**) x=10 y= x % 2 if y==0:  print(**'Even number'**)  if x>4:  print(**'x is greater than three'**)  else:  print(**'x is less than three'**) else:  print(**'Odd number'**)  in the console you will see:  all is well!  Even number  x is greater than three | x = 20  if x==10:  print(**"ten"**)  elif x==20:  print(**"twenty"**)  elif x==30:  print(**"thirty"**)  elif x==40:  print(**"forty"**)  else:  print(**"invalid input"**) |

Assignment:

Table : Assignment on if, elif, else and exit statement

|  |
| --- |
| Students shall select only one of the two Scenarios indicated in the boxes provided below and model the selected scenario in PyCharm environment and email. Students may use the given example below (solved example scenario). The example name.py file can also be found at this link ([Example\_Scenario](https://drive.google.com/file/d/1-6vpbBN7LtnIgrZN-ofDnodR-OGLskfo/view?usp=sharing)). ( <https://drive.google.com/file/d/1-6vpbBN7LtnIgrZN-ofDnodR-OGLskfo/view?usp=sharing> )  All scenarios indicated in this exercise file are extracted from an Indian GAT exams and modified for the exercise under consideration. |
| Example scenario: Students may use the following solved scenario as an example (take a look at the python code provided).   1. The Scenario to be used as an example for the exercise under consideration :   Eight people - A, B, C, D, E, F, G, and H – ride to work in three cars. Two cars each take three people, and one car takes only two people.  B rides with H.  G rides with only one other person.  F rides with two other people.  If C rides with B, the following pairs and triplets can ride together: (1) A and G; (2) G and E; (3) A, D and F and (4) B, C and H   1. The solution Python code to be used as an example is as follows:   Possible arrangement of people in the three cars can be: in car1=A, D, F; in car2=B, C, and H; in car3=G, E. The above order of cars is not the only possible order. |
| Answer example scenario : |
| *#1 (Given) # The Scenario to be used as an example for the exercise under consideration : #Eight people - A, B, C, D, E, F, G, and H – ride to work in three cars. # Two cars each take three people, and one car takes only two people. #B rides with H. #G rides with only one other person. #F rides with two other people #If C rides with B, the following pairs and triplets can ride together: # (1) A and G; (2) G and E; (3) A, D and F (4) B, C and H  #2 (Possible arrangement of people in the three cars can be:) #in car1 = A, D, F; #in car2 = B, C, H; #in car3 = G, E; #The above order is not the only possible order for the cars.  # The solution in Python code to be used as an example is as follows:*  start=input(**"Read the following Scenario. If you understand the Scenario and want to continue enter Y.**\n**"  "If you do not understand the scenario and do not want to continue enter N.**\n**"  "Eight people - A, B, C, D, E, F, G, and H – ride to work in three cars.**\n**"  "Two cars each take three people, and one car takes only two people.** \n**"  "B rides with H. G rides with only one other person. F rides with two other people.** \n**"  "If C rides with B, the following pairs and triplets can ride together:**\n**"  "(1) A and G; (2) G and E; (3) A, D and F and (4) B, C and H"**) if start ==**'Y'** or start==**'y'**:  car1 = input(**"Enter any person that can ride in car one: "**)  if car1 == **'A'** or car1==**'a'**:  print(**'Correct ! , the other people that can ride in this car are D and F'**)  elif car1==**'D'** or car1==**'d'**:  print(**'Correct ! , the other people that can ride in this car are A and F'**)  elif car1==**'F'** or car1==**'f'**:  print(**'Correct ! , the other people that can ride in this car are A and D'**)  elif car1==**'B'** or car1==**'b'** or car1==**'C'** or car1==**'c'** or car1==**'E'** or car1==**'e'** or car1==**'G'** or car1==**'g'** \  or car1==**'H'** or car1==**'h'**:  print(**'You are wrong !'**)  else:  print(**'Wrong input !'**)   car2 = input(**"Enter any person that can ride in car two: "**)  if car2 == **'B'** or car2==**'b'**:  print(**'Correct ! , the other people that can ride in this car are C and H'**)  elif car2==**'C'** or car2==**'c'**:  print(**'Correct ! , the other people that can ride in this car are B and H'**)  elif car2==**'H'** or car2==**'h'**:  print(**'Correct ! , the other people that can ride in this car are B and C'**)  elif car2==**'A'** or car2==**'a'** or car2==**'D'** or car2==**'d'** or car2==**'E'** or car2==**'e'** or car2==**'F'** or car2==**'f'** \  or car2==**'G'** or car2==**'g'**:  print(**'You are wrong'**)  else:  print(**'Wrong input !'**)   car3 = input(**"Enter any person that can ride in car three (Caps lock ): "**)  if car3 == **'G'** or car3==**'g'**:  print(**'Correct ! , the other person that can ride in this car is E'**)  elif car3==**'E'** or car3==**'e'**:  print(**'Correct ! , the other person that can ride in this car is G'**)  elif car3==**'A'** or car3==**'a'** or car3==**'B'** or car3==**'b'** or car3==**'C'** or car3==**'c'** or car3==**'D'** or car3==**'d'** \  or car3==**'F'** or car3==**'f'** or car3==**'H'** or car3==**'h'**:  print(**'You are wrong'**)  else:  print(**'Wrong input !'**)  elif start == **'N'** or start==**'n'**:  print(**"Thank you! Perhaps you can take a look at the Scenario next time and try again."**) else:  print(**"Wrong input, Good bye!"**)  finish=input(**"Please type E to exit or rerun the program to try again"**) if finish==**'E'** or finish==**'e'**:  exit else:  print(**"Wrong input, Good bye!"**)  exit |
| Select one of the following scenarios for your exercise and follow instruction given in the respective boxes. |
| Scenario option 1:  As a follow up to your previous class exercise, based on the following description of a specific scenario create a model in your PyCharm environment and email the resulting ‘name.py’ file to your instructor. Students should apply relevant information (to carry out the task under consideration) from all components of their covered course materials (python part 1, 2 and 3).  Six people – Ramzan, Shahid, Tayyab, Usman, Vajiha, and Waseem are standing in line for tickets to an upcoming concert.  Ramzan is fifth in line and is not next to Shahid.  Usman is immediately behind Tayyab.  Waseem is not last.  Vajiha is last in line. |
| Answer by students: |
| start=input(**"Six people – Ramzan, Shahid, Tayyab, Usman, Vajiha, and Waseem are standing in line for tickets to an upcoming concert.**\n**"  "Ramzan is fifth in line and is not next to Shahid.**\n**"  "Usman is immediately behind Tayyab.** \n**"  "Waseem is not last.** \n**"  "Vajiha is last in line.** \n**"  "The possible arrangement for the places are**\n**"  "(1st) Shahid, Tayyab, Waseem;** \n**"  "(2nd) Shahid, Waseem, Tayyab, (Usman if Tayyab is 1st)** \n**"  "(3rd) Shahid, Waseem, Tayyab, (Usman if Tayyab is 2nd)** \n**"  "(4th) Waseem, (Usman if Tayyab is 3rd); "  "(5th) Ramzan; "  "(6th) Vajiha**\n**"  "Enter Y to continue : "**) if start ==**'Y'** or start==**'y'**:  *#========================first line=======================#* first = input(**"Enter any person that can sit first : "**)  if first == **'Shahid'** or first==**'shahid'**:  print(**'Correct ! , the other people that can work on monday are Tayyab and Waseem '**)  elif first==**'Tayyab'** or first==**'tayyab'**:  print(**'Correct ! , the other people that can work on monday are Shahid and Waseem'**)  elif first==**'Waseem'** or first==**'waseem'**:  print(**'Correct ! , the other people that can work on monday are Shahid and Tayyab'**)  elif first==**'Usman'** or first==**'usman'** or first==**'Vajiha'** or first==**'vajiha'** or first==**'Ramzan'** or first==**'ramzan'**:  print(**'You are wrong ! the persons that can work on Monday are Shahid, Tayyab and Waseem'**)  else:  print(**'Wrong input !'**)  *#========================second line=======================#* second = input(**"Enter any person that can sit second: "**)  if first==**'Shahid'** or first == **'shahid'**:  if second == **'Shahid'** or second == **'shahid'**:  print(**'Shahid sit on first place'**)  if first != **'shahid'**and second == **'Shahid'** or first != **'shahid'**and second == **'shahid'** or\  first != **'shahid'**and second == **'Shahid'** or first != **'Shahid'**and second == **'Shahid'**:  print(**'Correct ! , the other people that sit second are Tayyab, Usman and Waseem'**)  elif second==**'Waseem'** or second==**'waseem'**:  print(**'Correct ! , the other people that sit second are Shahid, Tayyab and Usman'**)  elif first!= **'Tayyab'** and second==**'Tayyab'** or first!= **'tayyab'**and second==**'tayyab'** or first!= **'Tayyab'**and second==**'tayyab'** or \  first!= **'tayyab'**and second==**'Tayyab'** or second==**'Ramzan'** or second==**'ramzan'** or second==**'Vajiha'** or second==**'vajiha'**:  print(**'You are wrong'**)  else:  print(**'Wrong input !'**)   *# ========================third line=======================#* third = input(**"Enter any person that can sit third: "**)  if second==**'Shahid'** or second == **'shahid'**:  if third == **'Shahid'** or third == **'shahid'**:  print(**'Shahid sit on second place'**)  if second != **'shahid'**and third == **'Shahid'** or second != **'shahid'**and third == **'shahid'** or\  second != **'shahid'**and third == **'Shahid'** or second != **'Shahid'**and third == **'Shahid'**:  print(**'Correct ! , the other people that sit third are Tayyab and Waseem'**)  if first==**'Shahid'** or first == **'shahid'**:  if third == **'Shahid'** or third == **'shahid'**:  print(**'Shahid sit on first place'**)  if first != **'shahid'**and third == **'Shahid'** or first != **'shahid'**and third == **'shahid'** or\  first != **'shahid'**and third == **'Shahid'** or first != **'Shahid'**and third == **'Shahid'**:  print(**'Correct ! , the other people that sit third are Tayyab and Waseem'**)  elif third==**'Waseem'** or third==**'Waseem'**:  print(**'Correct ! , the other people that sit third are Shahid, Tayyab and Usman'**)  elif second!= **'Tayyab'** and third==**'Tayyab'** or second!= **'tayyab'**and third==**'tayyab'** or second!= **'Tayyab'**and third==**'tayyab'** or \  second!= **'tayyab'**and third==**'Tayyab'**:  print(**'Correct ! , the other people that sit third are Shahid and Waseem'**)  elif third==**'Ramzan'** or third==**'ramzan'** or third==**'Vajiha'** or third==**'vajiha'**:  print(**'You are wrong'**)  else:  print(**'Wrong input !'**)   *#========================fourth line=======================#* fourth = input(**"Enter any person that can sit fourth: "**)  if fourth==**'Waseem'** or fourth==**'waseem'**:  print(**'Correct ! , the other people that sit fourth is Usman'**)  if third == **'Tayyab'** or third == **'tayyab'**:  if fourth == **'Usman'** or fourth == **'usman'**:  print(**'Correct ! , the other people that sit fourth is Waseem'**)  elif third!= **'Tayyab'** and fourth==**'Usman'** or third!= **'tayyab'**and fourth==**'Usman'** or third!= **'Tayyab'**and fourth==**'usman'** or \  third!= **'tayyab'**and fourth==**'tayyab'** or fourth==**'Ramzan'** or fourth==**'ramzan'** or fourth==**'Vajiha'** or fourth==**'vajiha'**:  print(**'You are wrong'**)  else:  print(**'Wrong input !'**)   *#========================fifth line=======================#* fifth = input(**"Enter any person that can sit fifth: "**)  if fifth == **'Ramzan'** or fifth == **'ramzan'**:  print(**'Correct ! '**)  elif fifth == **'Usman'** or fifth == **'usman'** or fifth == **'Vajiha'** or fifth == **'vajiha'** or fifth == **'Waseem'** or fifth == **'waseem'**\  or fifth == **'Shahid'** or fifth == **'shahid'**or fifth == **'Tayyab'** or fifth == **'tayyab'**:  print(**'You are wrong'**)  else:  print(**'Wrong input !'**)  *#========================sixth last line=======================#* sixth = input(**"Enter any person that can sit sixth: "**)  if sixth == **'Varija'** or sixth == **'varija'**:  print(**'Correct ! '**)  elif sixth == **'Usman'** or sixth == **'usman'** or sixth == **'Ramzan'** or sixth == **'ramzan'** or sixth == **'Waseem'** or sixth == **'waseem'**\  or sixth == **'Shahid'** or sixth == **'shahid'**or sixth == **'Tayyab'** or sixth == **'tayyab'**:  print(**'You are wrong'**)  else:  print(**'Wrong input !'**)    elif start == **'N'** or start==**'n'**:  print(**"Thank you! Perhaps you can take a look at the Scenario next time and try again."**) else:  print(**"Wrong input, Good bye!"**)  finish=input(**"Please type E to exit or rerun the program to try again"**) if finish==**'E'** or finish==**'e'**:  exit else:  print(**"Wrong input, Good bye!"**)  exit |
| Scenario option 2:  As a follow up to your previous class exercise, based on the following description of a specific scenario create a model in your PyCharm environment and email the resulting ‘name.py’ file to your instructor. Students should apply relevant information (to carry out the task under consideration) from all components of their covered course materials (python part 1, 2 and 3).  The baby toy store employs five cashiers – Alam, Babar, Chohan, Dilawer, and Esa – each of whom works alone on exactly one day, Monday through Friday.  Alam will work only on Tuesday or Thursday.  Babar will not work on Monday or Wednesday.  Chohan works on Friday.  Dilawer and Esa do not work on consecutive days.  The following is a possible work schedule:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Monday | Tuesday | Wednesday | Thursday | Friday | | Esa | Alam | Dilawar | Babar | Chohan | |
| Answer by students: |
| start = input(**"Please read the scenario below and write Y/y if you understand the event or press N/n if you dont understand the event.** \n**" "The baby toy store employees five cashiers – Alam, Babar, Chohan, Dilawer, and Esa – each of whom** \n**" "works alone on exactly one day, Monday through Friday.** \n**" "Alam will work only on Tuesday or Thursday.** \n**" "Babar will not work on Monday or Wednesday.** \n**" "Chohan works on Friday.** \n**" "Dilawer and Esa do not work on consecutive days.** \n**" "You can not type one person twice since they all work only once a week!** \n**"  "type:" "A or a for Alam** \n**"  "B or b for Babar** \n**"  "C or c for Chohan** \n**"  "D or d for Dilawer** \n**"  "E or e for Esa"**)   if start == **"y"** or start == **"Y"**:  Monday = input(**"Enter the people that work on Mondays: "**)  if Monday == **"D"** or Monday == **"d"** or Monday == **"E"** or Monday == **"e"**:  print(**"Correct answer! That is a possible scenario on Monday"**)  elif Monday == **"A"** or Monday == **"a"** or Monday == **"B"** or Monday == **"B"** or Monday == **"C"** or Monday == **"c"**:  print(**"Wrong answer, he/she cannot work on Mondays"**)  else:  print(**"Wrong input!"**)   Tuesday= input(**"Enter the people that work on Tuesdays: "**)  if Tuesday == **"A"** or Tuesday == **"a"**:  print(**"Correct, A can work on Tuesdays"**)  elif Tuesday == **"B"** or Tuesday == **"b"**:  print(**"Correct, B can work on Tuesdays"**)  elif Tuesday == **"C"** or Tuesday == **"c"** or Tuesday == **"E"** or Tuesday == **"e"** or Tuesday == **"D"** or Tuesday == **"d"**:  print(**"Wrong answer, try again!"**)  else:  print(**"Wrong input"**)   Wednesday = input(**"Enter the people that work on Wednesday: "**)  if Wednesday == **"D"** or Wednesday == **"d"** or Wednesday == **"E"** or Wednesday == **"e"**:  print(**"Correct answer! That is a possible scenario"**)  elif Monday == **"d"** or Monday == **"D"**:  print(**"D already worked on Monday, a person con only work once a week!"**)  elif Wednesday == **"A"** or Wednesday == **"a"** or Wednesday == **"B"** or Wednesday == **"b"** or Wednesday == **"C"** or Wednesday == **"c"**:  print(**"Not correct! Please try another scenario!"**)  else:  print(**"Wrong input!"**)    Thursday = input(**"Enter the people that work on Thursdays: "**)  if Thursday == **"a"** or Thursday == **"A"**:  print(**"Correct, A can work on Thursdays"**)  elif Thursday == **"b"** or Thursday == **"B"**:  print(**"Correct, B can work on Thursdays"**)  elif Thursday == **"C"** or Thursday == **"c"** or Thursday == **"D"** or Thursday == **"d"** or Thursday == **"E"** or Thursday == **"e"**:  print(**"Your answer is not correct, try again"**)  else:  print(**"Wrong input"**)    Friday = input(**"Enter the people that work on Fridays: "**)  if Friday == **"C"** or Friday == **"c"**:  print(**"Correct, only C can work on Friday"**)  elif Friday == **"a"** or Friday == **"A"** or Friday == **"D"** or Friday == **"d"** or Friday == **"B"** or Friday == **"b"** or Friday == **"E"** or Friday == **"e"**:  print(**"Your answer is not correct, please try again"**)  else:  print(**"Wrong input"**)  elif start == **"N"** or start == **"N"**:  print(**"Thank you perhaps you can take another look and try again later!"**) else:  print(**"Wrong input!"**)  finish = input(**"Please type 'E' to exit the program or rerun the program to try again!"**) if finish == **"e"** or Finish == **"E"**:  print(**"Good Bye!"**) else:  print(**"Wrong input, goodbye"**) |
| Additional exercise: |
| Ethiopian tax Rate:   |  |  |  |  | | --- | --- | --- | --- | | 1. **No.** | 1. **Salary Range (ETB)** | 1. **Tax Rate** | 1. **Deduction (ETB)** | | 1. 1 | 1. 0 – 600 | 1. Non-Taxable | 1. – | | 1. 2 | 1. 601-1,650 | 1. 10% | 1. 60 | | 1. 3 | 1. 1,651 – 3,200 | 1. 15% | 1. 142.50 | | 1. 4 | 1. 3,201 – 5,250 | 1. 20% | 1. 302.50 | | 1. 5 | 1. 5,251 – 7,800 | 1. 25% | 1. 565 | | 1. 6 | 1. 7,801 – 10,900 | 1. 30% | 1. 955 | | 1. 7 | 1. Over 10,900 | 1. 35% | 1. 1,500 |      1. Based on the range aforementioned tax rate and range calculate tax of employee 2. calculate pension (7%) of each employee 3. calculate net pay of employee |
| Answer: |
| salary = int(input(**'please enter your salary: '**)) if 0 < salary < 600:  tax = **'Non Taxable salary'** pension = salary\*0.07  deduction = 0  net\_payment = salary  print (**'You should pay a tax amount of which is: '**, tax)  print (**'You have a pension value of: '**, pension)  print (**'Your salary has a deduction of: '**, deduction)  print (**'Your Net Payment is: '**, net\_payment) elif 601 < salary < 1650:  tax = salary\*0.1  pension = salary\*0.07  deduction = 60  net\_payment = salary - (tax - deduction) - pension  print (**'You should pay a tax amount of 10% which is: '**, tax)  print (**'You have a pension value of: '**, pension)  print (**'Your salary has a deduction of: '**, deduction)  print (**'Your Net Payment is: '**, net\_payment) elif 1651 < salary < 3200:  tax = salary\*0.15  pension = salary\*0.07  deduction = 142.5  net\_payment = salary - (tax - deduction) - pension  print (**'You should pay a tax amount of 15% which is: '**, tax)  print (**'You have a pension value of: '**, pension)  print (**'Your salary has a deduction of: '**, deduction)  print (**'Your Net Payment is: '**, net\_payment) elif 3201 < salary < 5250:  tax = salary\*0.2  pension = salary\*0.07  deduction = 302.5  net\_payment = salary - (tax - deduction) - pension  print (**'You should pay a tax amount of 20% which is: '**, tax)  print (**'You have a pension value of: '**, pension)  print (**'Your salary has a deduction of: '**, deduction)  print (**'Your Net Payment is: '**, net\_payment) elif 5251 < salary < 7800:  tax = salary\*0.25  pension = salary\*0.07  deduction = 565  net\_payment = salary - (tax - deduction) - pension  print (**'You should pay a tax amount of 25% which is: '**, tax)  print (**'You have a pension value of: '**, pension)  print (**'Your salary has a deduction of: '**, deduction)  print (**'Your Net Payment is: '**, net\_payment) elif 7801 < salary < 10900:  tax = salary\*0.3  pension = salary\*0.07  deduction = 955  net\_payment = salary - (tax - deduction) - pension  print (**'You should pay a tax amount of 30% which is: '**, tax)  print (**'You have a pension value of: '**, pension)  print (**'Your salary has a deduction of: '**, deduction)  print (**'Your Net Payment is: '**, net\_payment) elif salary > 10901:  tax = salary\*0.35  pension = salary\*0.07  deduction = 1500  net\_payment = salary - (tax - deduction) - pension  print (**'You should pay a tax amount of 35% which is: '**, tax)  print (**'You have a pension value of: '**, pension)  print (**'Your salary has a deduction of: '**, deduction)  print (**'Your Net Payment is: '**, net\_payment) |

# While loop and for loop in python

## While loops

While loop provides the programmer to run a function several times between an initial and provided condition. This is very handy in contrary to doing the output manually. The program needs the following components: Initialization, condition and increment (decrement).

Table : While loop

|  |  |
| --- | --- |
| x = 10  while x<20:  print (**"EiABC"**)  *#x=x+1* x+=1  Run this code in PyCharm and see what prints out. | x = 10 *#while x<20 and x>0 :* while x>1:  print (**"EiABC"**, 2000+x)  *#x=x+1* x-=1  Run this code in PyCharm and see what prints out. |
| x = 1  while x<10:  print (**"EiABC"**,end=**""**)  y=1  while y<=10:  print(**"Graduate students"**,end=**""**)  y=y+1  x=x+1  print()  Run this code in PyCharm and see what prints out. |  |

## For loops

While loop works with conditions while for loop works with sequences. As mentioned in your previous chapter (2.7.3. Sequence data types), sequences are concerned with: list, tuple, set, string, range.

Table : For loop

|  |  |
| --- | --- |
| x = [**'AAU'**, 2020, 12097] print(x[2])  list and index | x = [**'AAU'**, 2020, 12097] *#print(x[2])* for y in x:  print(y)  for loop in list |
| x = **'AAU'** *#print(x[2])* for y in x:  print(y)  for loop in a string | for y in [**'AAU'**, 2020, 12097]:  print(y)  for loop in list |
| for y in range(15):  print(y)  for loop in a string | for y in range(15,31,2):  print(y)  For loop in range with starting point, ending point and iteration. |
| for y in range(10,21):  if y%2==0:  print(y)  if inside a for loop with module moderator |  |

Exercise:

# Break, continue, pass, exit, try, except

## Break

The statement ‘break’ while working in loops allows to exit from the running function and jump to the next function in the program. Compare and contrast the different outputs in all three of the different cases in the following table:

|  |  |  |
| --- | --- | --- |
| Enter the value 3 in the console. | Enter the value 6 in the console. | Rearrange your code as follows and enter the value 3 and 6 in the console and see the result for each entry. |
| max = 5 x =int(input(**'How many people are there in town?'**)) i= 1 while i<=x:  i += 1  if x>max:  break  else:  print(**'People '**, end=**''**)  print(**' All is well'**) | max = 5 x =int(input(**'How many people are there in town?'**)) i= 1 while i<=x:  i += 1  if x>max:  break  else:  print(**'People '**, end=**''**)  print(**' All is well'**) | max = 5 x =int(input(**'How many people are there in town?'**)) i= 1 while i<=x:  i += 1  print(**'People '**, end=**''**)  if x > max:  break  print(**' All is well'**) |
| How many people are there in town?3  People People People All is well | How many people are there in town?6  All is well | How many people are there in town?3  People People People All is well  How many people are there in town?6  People All is well |

Edit the above code and see what prints in the console after entering different values as inputs:

|  |  |  |
| --- | --- | --- |
| Edit the above code as follows, run the program and enter the value 3 in the console | Edit the above code as follows, run the program and enter the value 6 in the console |  |
| max = 5 x =int(input(**'How many people are there in town?'**)) i= 1 while i<=x:  if i > max:  break  i += 1  print(**'People '**, end=**''**) | max = 5 x =int(input(**'How many people are there in town?'**)) i= 1 while i<=x:  if i > max:  break  i += 1  print(**'People '**, end=**''**) |  |
| How many people are there in town?3  People People People All is well | How many people are there in town?6  People People People People People All is well |  |

## Continue

While the statement ‘break’ allows to jump out of a loop (while or for), the statement ‘continue’ allows to omit a condition stated and continues with the loop without jumping out.

|  |  |  |
| --- | --- | --- |
|  | Compare the following codes with the exercises in the break statement examples above | |
| Enter the value 10 in the console | Enter the value 3 in the console |
| for i in range(1,51):  if i%2==0:  continue   print(i,end=**''**) | max = 5 x =int(input(**'How many people are there in town?'**)) i= 1 while i<=x:  if i > max:  continue  i += 1  print(**'People '**, end=**''**)   print(**' All is well'**) | max = 5 x =int(input(**'How many people are there in town?'**)) i= 1 while i<=x:  if i > max:  continue  i += 1  print(**'People '**, end=**''**)   print(**' All is well'**) |
| 1357911131517192123252  72931333537394143454749 | How many people are there in town?3  People People People All is well | How many people are there in town?6  People People People People People |

## Pass

The statement ‘pass’ allows a function to pass from a running function (block) to another block in a loop.

|  |  |
| --- | --- |
|  |  |
| for i in range (1,51):  if (i%3==0):  pass   else:  print(i ,end =**''**) |  |
| 12457810111314161719202223252  628293132343537384041434446474950 |  |

# Graphics user interface (GUI)

GUI is a visual way for users to interact with their computers or programmes. This include windows OS, webpages, etc. through different widgets like buttons, scrolls, boxes,etc.

## Different types of Graphic user interface (GUI) for python

There are different python GUI frameworks that we can use to create python GUI with. These frame works are all open source or have community or developers (commercial) packages. Among these the popular ones are: PYQT5 ( [www.riverbankcomputing.co.uk/news](http://www.riverbankcomputing.co.uk/news)), Tkinter (this framework is a built in to your python install and you do not need to install it separately), Wxpython (<http://wxpython.org>) , Kivy (the only solution to work with python code on mobile phones, it is mostly used for creating cross-platform multi-touch mobile applications) and PyForms.

## Tkinter (GUI)

We are using tkinter to work on exercises related to GUI.

Tkinter provides classes which allow the display, positioning and control of widgets. Toplevel widgets are Tk and Toplevel. Other widgets are Frame, Label, Entry, Text, Canvas, Button, Radiobutton, Checkbutton, Scale, Listbox, Scrollbar, OptionMenu, Spinbox, LabelFrame and PanedWindow.

Table : GUI general

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| GUI Tkinter component’s Exercises | | | | | |
| GUI Starts a window: | | GUI Grids | | | |
| *#starts a window that is 500x100*  from tkinter import\* *#imports modules related to GUI* root = Tk() *#creates the window* root.geometry(**"500x100"**) *#modifies root window* root.title(**"EIABC"**) *#Name of project* root.mainloop() *#puts the program into a long-running while loop called the event loop.puts the program into a long-running while loop called the event loop.* | | *#creates a 2 by 2 grid* from tkinter import \*  root= Tk() root.title(**"EIABC\_Grid"**)  label1 = Label(text=**'Lable1'**) label1.grid(row=0, column=0) label2 = Label(text=**'Lable2'**) label2.grid(row=0, column=1) label3 = Label(text=**'Lable3'**) label3.grid(row=1, column=0) label4 = Label(text=**'Lable4'**) label4.grid(row=1, column=1)  mainloop() | | | |
| GUI Label | | GUI Grids (Grid arrangement of the GUI (both column and row start from 0) | | | |
| *# Creates lebel*  from tkinter import \*  root= Tk()  hello\_label = Label(text=**'hello'**) hello\_label.grid(row=0, column=1)  mainloop() | | Table : GUI Grids   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | *0,0* | *0,1* | *0,2* | *0,3* | *0,4* | *0,5* | *0,6* | *(N rows, N columns)* | | *1,0* | *1,1* | *1,2* | *1,3* | *1,4* | *1,5* | *1,6* | *(N rows, N columns* | | *2,0* | *2,1* | *2,2* | *2,3* | *2,4* | *2,5* | *2,6* | *(N rows, N columns* | | *3,0* | *3,1* | *3,2* | *3,3* | *3,4* | *3,5* | *3,6* | *(N rows, N columns* | | *4,0* | *4,1* | *4,2* | *4,3* | *4,4* | *4,5* | *4,6* | *(N rows, N columns* | | *5,0* | *5,1* | *5,2* | *5,3* | *5,4* | *5,5* | *5,6* | *(N rows, N columns* | | | | |
| GUI Entry boxes | GUI Buttons | | | | GUI List |
| *# Creates Entry boxes*  from tkinter import \*  root= Tk() root.title(**"EIABC\_Entry Boxes"**)  entry = Entry() entry.grid(row=0, column=0) entry.insert(0, **'hello, enter value here!'**)  mainloop() | *# Creates Buttons*  from tkinter import \*  def callback(): label.configure(text=**'Button clicked'**)  root = Tk() root.title(**"EIABC\_Buttons"**)  label = Label(text=**'Not clicked'**) button = Button(text=**'Click me'**, command=callback) label.grid(row=0, column=0) button.grid(row=1, column=0)  mainloop() | | | | *# Creates list*  from tkinter import \*  root = Tk() root.title(**"EIABC\_List"**)  listbox = Listbox(root) listbox.pack() listbox.insert(END, **"a list entry"**) for item in [**"one"**, **"two"**, **"three"**, **"four"**]:  listbox.insert(END, item)  mainloop() |
| GUI Scrollbars | GUI Checkbox | | | | GUI Option button |
| *# creates scrollbars* from tkinter import \*  root = Tk() root.title(**"EIABC\_Scrollbar"**)  root.title(**'scroll bar'**) root.geometry(**'150x150'**) w = Scale(root, from\_=0, to=42) w.pack() w = Scale(root, from\_=0, to=200, orient=HORIZONTAL) w.pack()  mainloop() | *# Creates checkbox* from tkinter import \*  root = Tk() root.title(**'EiABC checkbox'**) root.geometry(**'200x200'**)  CheckVar1 = IntVar() CheckVar2 = IntVar() C1 = Checkbutton(root, text = **"Music"**, variable = CheckVar1, \ onvalue = 1, offvalue = 0, height=5, \ width = 20, ) C2 = Checkbutton(root, text = **"Video"**, variable = CheckVar2, \ onvalue = 1, offvalue = 0, height=5, \ width = 20) C1.pack() C2.pack()  root.mainloop() | | | | *#Creates option button* from tkinter import \*  def sel():  selection = **"You selected the option "** + str(var.get())  label.config(text = selection)  root = Tk() root.title(**'EiABC option button'**)  root.geometry(**'300x100'**) var = IntVar() R1 = Radiobutton(root, text=**"Option 1"**, variable=var, value=1, command=sel) R1.pack( anchor = W ) R2 = Radiobutton(root, text=**"Option 2"**, variable=var, value=2, command=sel) R2.pack( anchor = W ) R3 = Radiobutton(root, text=**"Option 3"**, variable=var, value=3, command=sel) R3.pack( anchor = W) label = Label(root) label.pack()  root.mainloop() |
| GUI Canvas | | | | | |
| Lines | | | | Lines, fills and colors | |
| from tkinter import Tk, Canvas, Frame, BOTH  class Example(Frame):  def \_\_init\_\_(self):  super().\_\_init\_\_()  self.initUI()  def initUI(self):  self.master.title(**"EIABC\_Canvas1\_Lines"**)  self.pack(fill=BOTH, expand=1)  canvas = Canvas(self)  canvas.create\_line(15, 25, 200, 25)  canvas.create\_line(300, 35, 300, 200, dash=(4, 2))  canvas.create\_line(55, 85, 155, 85, 105, 180, 55, 85)  canvas.pack(fill=BOTH, expand=1) def main():   root = Tk()  ex = Example()  root.geometry(**"400x250+300+300"**)   root.mainloop()  if \_\_name\_\_ == **'\_\_main\_\_'**:  main() | | | | from tkinter import Tk, Canvas, Frame, BOTH class Example(Frame):  def \_\_init\_\_(self):  super().\_\_init\_\_()  self.initUI()  def initUI(self):  self.master.title(**"EIABC\_Canvas2\_Lines\_Fill\_Colours"**)  self.pack(fill=BOTH, expand=1)  canvas = Canvas(self)  canvas.create\_rectangle(30, 10, 120, 80, outline=**"#fb0"**, fill=**"#fb0"**)  canvas.create\_rectangle(150, 10, 240, 80, outline=**"#f50"**, fill=**"#f50"**)  canvas.create\_rectangle(270, 10, 370, 80, outline=**"#05f"**, fill=**"#05f"**)  canvas.pack(fill=BOTH, expand=1) def main():  root = Tk()  ex = Example()  root.geometry(**"400x100+300+300"**)  root.mainloop()  if \_\_name\_\_ == **'\_\_main\_\_'**:  main() | |
| GUI Lambda | | | | | |
| *#Uses one function for multiple events.* from tkinter import \*  alphabet = **'ABCDEFGHIJKLMNOPQRSTUVWXYZ'** def callback(x):  label.configure(text=**'Button {} clicked'**.format(alphabet[x]))  root = Tk() root.title(**"EIABC\_Lambda"**)  label = Label() label.grid(row=1, column=0, columnspan=26) buttons = [0]\*26 *# create a list to hold 26 buttons* for i in range(26):  buttons[i] = Button(text=alphabet[i],  command = lambda x=i: callback(x))  buttons[i].grid(row=0, column=i) | | | | | |
| GUI Text and time | | | | | |
| from tkinter import\* import time; *# imports module related to time* root = Tk() root.geometry(**"500x100"**) root.title(**"EIABC"**) text\_Input=StringVar() *#declaration for text input variable* operator=(**""**) x = Frame(root, width=500 ,height=200,bg=**"gray"**)x.pack(side=TOP) localtime=time.asctime(time.localtime(time.time())) label = Label(x, font=(**'areal'**, 35, **'bold'**), text=**"EIABC"**, fg=**"red"**) label.grid(row=0, column=0) label= Label(x, font=(**'areal'**, 20, **'bold'**), text=localtime, fg=**"black"**) label.grid(row=5, column=0) root.mainloop() | | | | | |
| *GUI Class* exercises*: Grade* | | | | | |
| GUI grade | | | | | |
| *Create a program for computing grade of students by making use of (a) if and else (b) if, elif and else. And create a GUI for any one of the two options, a or b. Use the following rules to simulate the task:: (1) students that score marks from 90 to 100 score are graded with ‘A’, from 75 to 90 with ‘B’, from 45 to 75 with ‘C’, from 30 to 44 with ‘D’, from 30 to 0 with ‘F’. (2) if students score any mark different from the above ranges, they are scored with ‘wrong score’* | | | | | |
| *Answer:* | | | | | |
| *Step one: solve the problem with (a) if and (b) elif and else* | | | | | |
| *(a)If* | | | *(b) elif and else* | | |
| *#evaluates grade through bool function* from tkinter import \*  def bool():   score = float(entry\_score.get())  *#score= float(input ("Enter score:"))* if score >= 90 and score <= 100:  grade = **'A'** elif score >= 70 and score < 90:  grade = **'B'** elif score >= 45 and score < 70:  grade = **'C'** elif score >= 30 and score < 45:  grade = **'D'** elif score >= 0 and score < 30:  grade = **'F'** else:  grade = **"wrong score"** output\_label\_grade.configure(text=**'Grade of the Student is: {}'**.format(grade))  root = Tk() root.title(**"EIABC\_Grade"**)  inquiry\_label\_score = Label(text=**'Enter Score of the Student'**, font=(**'Broadway'**, 16), fg=**'black'**) inquiry\_label\_score.grid(row=0, column=0)  entry\_score = Entry(font=(**'Tekton Pro'**, 16), width=9) entry\_score.insert(0, **'?!'**) entry\_score.grid(row=1, column=0)  output\_label\_grade = Label(font=(**'Bauhaus 93'**, 16), fg=**'red'**) eval\_button\_grade = Button(text=**'Evaluate'**, font=(**'Verdana'**, 16), command=bool )  eval\_button\_grade.grid(row=2, column=0) output\_label\_grade.grid(row=3, column=0)  mainloop() | | | *#evaluates grade through bool function* from tkinter import \*  def bool():   score = float(entry\_score.get())  *#score= float(input ("Enter score:"))* if score >= 90 and score <= 100:  grade = **'A'** if score >= 70 and score < 90:  grade = **'B'** if score >= 45 and score < 70:  grade = **'C'** if score >= 30 and score < 45:  grade = **'D'** if score >= 0 and score < 30:  grade = **'F'** else:  grade = **"wrong score"** output\_label\_grade.configure(text=**'Grade of the Student is: {}'**.format(grade))  root = Tk() root.title(**"EIABC\_Grade"**)  inquiry\_label\_score = Label(text=**'Enter Score of the Student'**, font=(**'Broadway'**, 16), fg=**'black'**) inquiry\_label\_score.grid(row=0, column=0)  entry\_score = Entry(font=(**'Tekton Pro'**, 16), width=9) entry\_score.insert(0, **'?!'**) entry\_score.grid(row=1, column=0)  output\_label\_grade = Label(font=(**'Bauhaus 93'**, 16), fg=**'red'**) eval\_button\_grade = Button(text=**'Evaluate'**, font=(**'Verdana'**, 16), command=bool )  eval\_button\_grade.grid(row=2, column=0) output\_label\_grade.grid(row=3, column=0)  mainloop() | | |
| Figure : Grades | | | | | |
| More exercises: | | | | | |
| 1. Based on the following grade range calculate grade using calculate button   |  |  | | --- | --- | | **Raw Mark Interval** | **Corresponding Grade** | | >100 and < 1 | Wrong grade | | 90-100 | A+ | | 83-90 | A | | 80-83 | A– | | 75- 80 | B+ | | 68- 75 | B | | 65- 68 | B– | | 60-65 | C+ | | 50- 60 | C | | 45- 50 | C– | | 40-45 | D | | 30-40 | Fx | | <30 | F |   2. Create first Name, Last Name, Gender, Mark and Grade  3. Create Check Box Male and Female  4. Window title must be Grade calculator | | | | | |
| Answer by students: | | | | | |
| from tkinter import\*  def cal():  grade=int(entry.get())  firstname=str(finame.get())  lastname=str(laname.get())  *#entry.delete(0,END)* if grade > 100:  text4.configure(text=**'Welcome '**+ firstname + **" "** ,bg=**"gray"**)  text7.configure(text=**'Name: '**+ firstname + **" "** + lastname ,bg=**"goldenrod"**)  text6.configure(text=**"Gender: "**+ var1.get(),bg=**"greenyellow"**)  text5.configure(text=firstname+**" "** +**'Wrong input, make sure you Entered the right value'** ,bg=**"red"**,fg=**"white"**)   elif 90 <= grade < 100:  text4.configure(text=**'Welcome '**+ firstname + **" "** + lastname ,bg=**"gray"**)  text7.configure(text=**'Name: '**+ firstname + **" "** + lastname ,bg=**"goldenrod"**)  text6.configure(text=**"Gender: "**+ var1.get(),bg=**"greenyellow"**)  text5.configure(text=**'Your Grade is: A+'**,bg=**"brown"**)     elif 83 <= grade < 90:  text4.configure(text=**'Welcome '**+ firstname + **" "** + lastname ,bg=**"gray"**)  text7.configure(text=**'Name '**+ firstname + **" "** + lastname ,bg=**"goldenrod"**)  text6.configure(text=**"Gender: "**+ var1.get(),bg=**"greenyellow"**)  text5.configure(text=**'Your Grade is: A'**,bg=**"brown"**)     elif 80 <= grade < 83:  text4.configure(text=**'Welcome '**+ firstname + **" "** + lastname ,bg=**"gray"**)  text7.configure(text=**'Name '**+ firstname + **" "** + lastname ,bg=**"goldenrod"**)  text6.configure(text=**"Gender: "**+ var1.get(),bg=**"greenyellow"**)  text5.configure(text=**'Your Grade is: A- '**,bg=**"brown"**)  window.geometry(**"600x500"**)    elif 75 <= grade < 80:  text4.configure(text=**'Welcome '**+ firstname + **" "** + lastname ,bg=**"gray"**)  text7.configure(text=**'Name '**+ firstname + **" "** + lastname ,bg=**"goldenrod"**)  text5.configure(text=**'Your Grade is: B+ '**,bg=**"brown"**)  text6.configure(text=**"Gender: "**+ var1.get(),bg=**"greenyellow"**)    elif 68 <= grade < 75:  text4.configure(text=**'Welcome '**+ firstname + **" "** + lastname ,bg=**"gray"**)  text7.configure(text=**'Name '**+ firstname + **" "** + lastname ,bg=**"goldenrod"**)  text6.configure(text=**"Gender: "**+ var1.get(),bg=**"greenyellow"**)  text5.configure(text=**'Your Grade is B '**,bg=**"brown"**)      elif 65 <= grade < 68:  text4.configure(text=**'Welcome '**+ firstname + **" "** + lastname ,bg=**"gray"**)  text7.configure(text=**'Name '**+ firstname + **" "** + lastname ,bg=**"goldenrod"**)  text6.configure(text=**"Gender: "**+ var1.get(),bg=**"greenyellow"**)  text5.configure(text=**'Your Grade is B-'**,bg=**"brown"**)    elif 60 <= grade < 65:  text4.configure(text=**'Welcome '**+ firstname + **" "** + lastname ,bg=**"gray"**)  text7.configure(text=**'Name '**+ firstname + **" "** + lastname ,bg=**"goldenrod"**)  text5.configure(text=**'Your Grade is: C+ '**,bg=**"brown"**)  text6.configure(text=**"Gender: "**+ var1.get(),bg=**"greenyellow"**)  window.geometry(**"600x500"**)  elif 50 <= grade < 60:  text4.configure(text=**'Welcome '**+ firstname + **" "** + lastname ,bg=**"gray"**)  text7.configure(text=**'Name '**+ firstname + **" "** + lastname ,bg=**"goldenrod"**)  text6.configure(text=**"Gender "**+ var1.get(),bg=**"greenyellow"**)  text5.configure(text=**'Your Grade is: C '**,bg=**"brown"**)     elif 45 <= grade < 50:  text4.configure(text=**'Welcome '**+ firstname + **" "** + lastname ,bg=**"gray"**)  text7.configure(text=**'Name '**+ firstname + **" "** + lastname ,bg=**"blue"**)  text6.configure(text=**"Gender "**+ var1.get(),bg=**"greenyellow"**)  text5.configure(text=**'Your Grade is: C- '**,bg=**"brown"**)     elif 40 <= grade < 45:  text4.configure(text=**'Welcome '**+ firstname + **" "** + lastname ,bg=**"gray"**)  text7.configure(text=**'Name '**+ firstname + **" "** + lastname ,bg=**"goldenrod"**)  text6.configure(text=**"Gender: "**+ var1.get(),bg=**"greenyellow"**)  text5.configure(text=**'Your Grade is: D '**,bg=**"brown"**)      elif 30 <= grade < 40:  text4.configure(text=**'Welcome '**+ firstname + **" "** + lastname ,bg=**"gray"**)  text7.configure(text=**'Name '**+ firstname + **" "** + lastname ,bg=**"goldenrod"**)  text6.configure(text=**"Gender: "**+ var1.get(),bg=**"greenyellow"**)  text5.configure(text=**'Your Grade is: Fx '**,bg=**"brown"**)      elif 0<= grade < 30:  text4.configure(text=**'Welcome '**+ firstname + **" "** + lastname ,bg=**"gray"**)  text7.configure(text=**'Name '**+ firstname + **" "** + lastname ,bg=**"goldenrod"**)  text6.configure(text=**"Gender: "**+ var1.get(),bg=**"greenyellow"**)  text5.configure(text=**'Your Grade is: F '**,bg=**"brown"**)  window.geometry(**"600x500"**)    elif grade < 0:  text4.configure(text=**'Welcome '**+ firstname + **" "** + lastname ,bg=**"gray"**)  text7.configure(text=**'Name '**+ firstname + **" "** + lastname ,bg=**"goldenrod"**)  text6.configure(text=**"Gender: "**+ var1.get(),bg=**"greenyellow"**)  text5.configure(text=firstname+**" "** +**'I Think You Enterd Wrong input, make sure you Entered the right value'**,bg=**"red"**,fg=**"white"**)       window=Tk() window.title(**"Grade Calculation"**) window.configure(background=**"gray"**) window.geometry(**"600x500"**)  text1=Label(window, text=**"Enter Your Result Here: "**, width=25,font=(**"Calibri"**,11)) text1.grid(row=7, column=2,sticky=W,pady=10,padx=10)  entry=Entry(window, width=20, bg=**"white"**, fg=**"black"**, font=(**"Calibri"**,10)) entry.grid(row=7, column=3,sticky=W,pady=10,padx=10)  btn1=Button(window, text=**"Calculate"**, bg=**"darkblue"**, fg=**"white"**, command=cal, font=(**"Calibri"**,13)) btn1.grid(row=15, column=3, sticky=W,pady=10)    fname=Label(window, text=**"Enter Your First Name "**, width=25,font=(**"Calibri"**,10), bg=**"white"** ,fg=**"black"** ) fname.grid(row=3, column=2,sticky=W,pady=10,padx=10) finame=Entry(window,width=20, bg=**"white"**, fg=**"black"**, font=(**"Calibri"**,9)) finame.grid(row=3, column=3,sticky=W,pady=10,padx=10)  lname=Label(window, text=**"Enter Your Last Name "** ,width=25,font=(**"Calibri"**,10), bg=**"white"** , fg=**"black"**) lname.grid(row=5, column=2 ,sticky=W,pady=10,padx=10) laname=Entry(window,width=20, bg=**"white"**, fg=**"black"**, font=(**"Calibri"**,9)) laname.grid(row=5, column=3,sticky=W,pady=10,padx=10)  gender=Label(window, text=**"Select Your Gender "**,width=25,font=(**"Calibri"**,10), bg=**"white"** , fg=**"black"** ) gender.grid(row=6, column=2,sticky=W, pady=10,padx=10)  var1 = StringVar()  c1 = Radiobutton(window, text = **"Male"**, variable = var1,value = **"Male"**, height=2, width = 6, ) c1.deselect() c1.grid(row=6, column=3,sticky=W,pady=10,padx=10)  c2 = Radiobutton(window, text = **"Female"**, variable = var1,value = **"Female"**, height=2,width =6) c2.grid(row=6, column=4,sticky=W,pady=10)  text4=Label(window, font=(**"Calibri"**,9), bg=**"greenyellow"** , fg=**"black"**) text4.grid(row=9, column=2,sticky=W, pady=10,padx=10)  text5=Label(window, font=(**"Calibri"**,9), bg=**"brown"**, fg=**"black"**) text5.grid(row=12, column=2,sticky=W, pady=10,padx=10)  text6=Label(window, font=(**"Calibri"**, 9)) text6.grid(row=11, column=2,sticky=W ,pady=10,padx=10)  text7=Label(window, font=(**"Calibri"**, 9)) text7.grid(row=10, column=2,sticky=W ,pady=10,padx=10)  window.mainloop() | | | | | |
| Alternative answer by students: | | | | | |
| from tkinter import \*  root = Tk() root.title(**"Grade Calculator"**) root.geometry(**"700x500"**)   def calcGrade(grade):  if (grade > 100 or grade < 1):  return **"Wrong grade"** elif (grade >= 90 and grade <= 100):  return **"A+"** elif (grade >= 83 and grade < 90):  return **"A"** elif (grade >= 80 and grade < 83):  return **"A-"** elif (grade >= 75 and grade < 80):  return **"B+"** elif (grade >= 68 and grade < 75):  return **"B"** elif (grade >= 65 and grade < 68):  return **"B-"** elif (grade >= 60 and grade < 65):  return **"C+"** elif (grade >= 50 and grade < 60):  return **"C"** elif (grade >= 45 and grade < 50):  return **"C-"** elif (grade >= 40 and grade < 45):  return **"D"** elif (grade >= 30 and grade < 40):  return **"Fx"** elif (grade < 30):  return **"F"** else:  return **"Invald Input"** label\_Firstname = Label(root, text=**"First Name "**).pack() Firstname = Entry(root).pack() label\_Lastname = Label(root, text=**"Last Name "**).pack() Lastname = Entry(root).pack() label\_Gender = Label(root, text=**"Gender "**).pack() var1 = IntVar() c = Checkbutton(root, text=**"Male"**, variable=var1,  onvalue=1, offvalue=0, height=3, width=10, ).pack() var2 = IntVar() c2 = Checkbutton(root, text=**"Female"**, variable=var2,  onvalue=1, offvalue=0, height=3,  width=10).pack() label\_Grade = Label(root, text=**"Grade "**).pack()  grade\_var = StringVar() Grade = Entry(root, textvariable=grade\_var).pack()  gradeResult = StringVar()   def click():  gradeResult.set(calcGrade(eval(grade\_var.get())))   Btn = Button(root, text=**"Calculate"**, command=click).pack(pady=1)  result = Label(root, text=**"Result"**).pack(pady=10) label\_Result = Label(root, textvariable=gradeResult).pack()  mainloop() | | | | | |
| Alternative answer by students: | | | | | |
| from tkinter import \*  root = Tk() *##Commands for the labels in the 0th column* firstNameLabel = Label(text = **'First name'**, font = (**'verdana'**, 10, **'bold'**), bg = **'Red'**, fg = **'green'**) firstNameLabel.grid(row = 0, column = 0) lastNameLabel = Label(text = **'Last name'**) lastNameLabel.grid(row = 1, column = 0) sexLabel = Label(text = **'Sex'**) sexLabel.grid(row = 2, column = 0) markLabel = Label(text = **'Mark'**) markLabel.grid(row = 3, column = 0) gradeLabel = Label(text = **'Grade'**) gradeLabel.grid(row = 4, column = 0) listLabel = Label(text = **'info'**) listLabel.grid(row = 5, column = 0) *##Commands for the entries in the 1st column* firstNameEntry = Entry(root,width=50) firstNameEntry.grid(row=0,column=1) lastNameEntry=Entry(root,width=50) lastNameEntry.grid(row=1,column=1) sexEntry=Entry(root,width=50) sexEntry.grid(row=2,column=1) markEntry=Entry(root,width=50) markEntry.grid(row=3,column=1)    def call():  x = list()  x.append(firstNameEntry.get())  x.append(lastNameEntry.get())  x.append(sexEntry.get())  print (x)  def myClick():  try:  compute\_grade(float(markEntry.get()))  except:  print(**"Please input a correct value!"**) def compute\_grade(score):  global value  try:  if 90 <= score <=100:  value = 4  print(**'A+ '**, **'fixed number grade is:'**, value)  gradeValue = Label(text = **'A+'** + **' fixed number grade is: 4'**)  gradeValue.grid(row = 4, column = 1)   elif 83 <= score < 90:  value = 4  print(**'A '**, **'fixed number grade is:'**, value)  gradeValue = Label(text = **'A '** + **'fixed number grade is: 4'**)  gradeValue.grid(row = 4, column = 1)    elif 80 <= score < 83:  value = 3.75  print(**'A-'**, **'fixed number grade is:'**, value)  gradeValue = Label(text = **'A- '** + **'fixed number grade is: 3.75'**)  gradeValue.grid(row = 4, column = 1)   elif 75 <= score < 80:  value = 3.5  print(**'B+'**, **'fixed number grade is:'**, value)  gradeValue = Label(text = **'B+ '** + **'fixed number grade is: 3.5'**)  gradeValue.grid(row = 4, column = 1)   elif 68 <= score < 75:  value = 3  print(**'B'**, **'fixed number grade is:'**, value)  gradeValue = Label(text = **'B '** + **'fixed number grade is: 3'**)  gradeValue.grid(row = 4, column = 1)   elif 65 <= score < 68:  value = 2.75  print(**'B-'**, **'fixed number grade is:'**, value)  gradeValue = Label(text = **'B- '** + **'fixed number grade is: 2.75'**)  gradeValue.grid(row = 4, column = 1)   elif 60 <= score < 65:  value = 2.5  print(**'C+'**, **'fixed number grade is:'**, value)  gradeValue = Label(text = **'C+ '** + **'fixed number grade is: 2.5'**)  gradeValue.grid(row = 4, column = 1)   elif 50 <= score < 60:  value = 2  print(**'C'**, **'fixed number grade is:'**, value)  gradeValue = Label(text = **'C '** + **'fixed number grade is: 2'**)  gradeValue.grid(row = 4, column = 1)   elif 45 <= score < 50:  value = 1.75  print(**'C-'**, **'fixed number grade is:'**, value)  gradeValue = Label(text = **'C- '** + **'fixed number grade is: 1.75'**)  gradeValue.grid(row = 4, column = 1)   elif 40 <= score < 45:  value = 1  print(**'D'**, **'fixed number grade is:'**, value)  gradeValue = Label(text = **'D '** + **'fixed number grade is: 1'**)  gradeValue.grid(row = 4, column = 1)   elif 30 <= score < 40:  value = **'undetermined'** print(**'FX'**, **'fixed number grade is:'**, value)  gradeValue = Label(text = **'FX '** + **'fixed number grade is: undetermined'**)  gradeValue.grid(row = 4, column = 1)   elif score < 30:  value = 0  print(**'F'**, **'fixed number grade is:'**, value)  gradeValue = Label(text = **'F '** + **'fixed number grade is: 0'**)  gradeValue.grid(row = 4, column = 1)  except:  return  myButton = Button(root,text=**"Go!"**,padx=50,command=myClick,bg=**"Red"**,fg=**"Blue"**) myButton.grid(row=2,column=3)  root.mainloop() | | | | | |
| Exercise (GUI): Exponential growth | | | | | |
| In your previous class you have simulated exponential salary increase of an employee with the equation y=a(r+1)t by making use of input function and the console of your python program.  t= number of years of experience before salary of employee gets to y amount.  r=rate of salary increment in a year.  y=salary of an employee in t years.    For this exercise construct a graphics user interface (GUI) for the same formula.  Hint:  Your GUI should have label and input for a, r, t.  Your GUI should have label and output for y.  Your GUI should have calculate or compute button.  Students can choose their own fg and bg colors of their own as long as all buttons, outputs and inputs are visible. | | | | | |
| Answer: | | | | | |
| import math  from tkinter import \*  def calculate():  t=int(entry\_t.get())  a=float(entry\_a.get())  r= float(entry\_r.get())  sy =a\*pow((r+1),t)  output\_label\_sy.configure(text=**'Salary After Y Years: {}'**.format(sy))   *#entry.delete(0,END)* root = Tk() root.title(**"EIABC\_Salary"**)  inquiry\_label\_t = Label(text=**'Enter Number of Experience in Years'**, font=(**'Broadway'**, 16), fg=**'black'**) inquiry\_label\_t.grid(row=0, column=0) entry\_t = Entry(font=(**'Tekton Pro'**, 16), width=4) entry\_t.insert(0, **'?!'**) entry\_t.grid(row=0, column=1)  inquiry\_label\_a = Label(text=**'Enter Basic Salary'**, font=(**'Broadway'**, 16,), fg=**'black'**) inquiry\_label\_a.grid(row=1, column=0) entry\_a = Entry(font=(**'Tekton Pro'**, 16), width=4) entry\_a.insert(0, **'?!'**) entry\_a.grid(row=1, column=1)  inquiry\_label\_r = Label(text=**'Enter Rate of Salary Increment'**, font=(**'Broadway'**, 16,),fg=**'black'**) inquiry\_label\_r.grid(row=2, column=0) entry\_r = Entry(font=(**'Tekton Pro'**, 16), width=4) entry\_r.insert(0, **'?!'**) entry\_r.grid(row=2, column=1)  output\_label\_sy = Label(font=(**'Bauhaus 93'**, 16), fg=**'red'**) calc\_button\_sy = Button(text=**'Calculate'**, font=(**'Verdana'**, 16), command=calculate)  calc\_button\_sy.grid(row=0, column=2) output\_label\_sy.grid(row=3, column=0, columnspan=2)  mainloop() | | | | | |
| *Sample imputes and outputs: (1) input(Number of years=10, Basic salary=50, rate=0.3), (output=68929.245) (2) input(Number of years=10, Basic salary=1000, rate=0.35), (output=201065.558)* | | | | | |
| *The GUI with three columns and four rows:*    Figure : Exponential Growth 01 | | | | | |
| *Change the above GUI As follows, with a single column, and 8 rows.* | | | | | |
| Figure : Exponential growth 02 | | | | | |
| GUI temperature convert | | | | | |
| *Create a GUI to convert Fahrenheit Temperature to Celsius: Celsius = (9/5)\*Fahrenheit+32* | | | | | |
| *Answer* | | | | | |
| *# the following programme converts temperatures from Fahrenheit to Celsius.*  *#Celsius = (9/5)\*Fahrenheit+32* from tkinter import \*  def calculate():  temp = int(entry.get())  temp = 9/5\*temp+32  output\_label.configure(text = **'Converted in Degree Celsius: {:.1f}'**.format(temp))  *#entry.delete(0,END)* root = Tk() root.title(**"EIABC\_Temp\_Convert:Fahrenheit to Celsius"**)  inquiry\_label = Label(text=**'Enter a temperature in Fahrenheit'**, font=(**'Broadway'**, 16,),bg=**'black'**, fg=**'white'**) output\_label = Label(font=(**'Bauhaus 93'**, 16), bg=**'black'**, fg=**'white'**)  entry = Entry(font=(**'Tekton Pro'**, 16), width=4) entry.insert(0, **'?!'**)  calc\_button = Button(text=**'Ok'**, font=(**'Verdana'**, 16), command=calculate)  inquiry\_label.grid(row=0, column=0) entry.grid(row=0, column=1) calc\_button.grid(row=0, column=2) output\_label.grid(row=1, column=0, columnspan=2)  mainloop() | | | | | |
| Figure : Temp convert | | | | | |
| Calculator: | | | | | |
| from tkinter import\* *#imported modules related to GUI  #* root = Tk() *#creates the window* root.geometry(**"500x600"**) *#modifies root window* root.title(**"EIABC"**) *#Name of project #* text\_Input=StringVar() *#declaration for text input variable* operator=(**""**) *#* f2 = Frame(root, width=500 ,height=500,bg=**"yellow"**) *#modifies root Frame )* f2.pack(side=BOTTOM) *#  #* def btnClick(numbers):  global operator  operator=operator + str(numbers)  text\_Input.set(operator) def btnClearDisply():  global operator  operator=**""** text\_Input.set(**""**) def btnEqualsInput():  global operator  sumup=str(eval(operator))  text\_Input.set(sumup)  operator=**""** *#* textDisplay = Entry (f2,font=(**'arial'**, 20, **'bold'**), textvariable=text\_Input,bd=30, insertwidth=4,  bg=**"white"**, justify=**'right'**) textDisplay.grid(columnspan=5) *#* btn7=Button(f2,padx=16, pady=16, bd=8, fg=**"black"**, font=(**'arial'**,20,**'bold'**),  text=**"7"**, bg=**"yellow"**, command=lambda: btnClick(7)).grid(row=2, column=0) btn8=Button(f2,padx=16, pady=16, bd=8, fg=**"black"**, font=(**'arial'**,20,**'bold'**),  text=**"8"**, bg=**"yellow"**, command=lambda: btnClick(8)).grid(row=2, column=1) btn9=Button(f2,padx=16, pady=16, bd=8, fg=**"black"**, font=(**'arial'**,20,**'bold'**),  text=**"9"**, bg=**"yellow"**, command=lambda: btnClick(9)).grid(row=2, column=2) *#* Addition=Button(f2,padx=16, pady=16, bd=8, fg=**"black"**, font=(**'arial'**,20,**'bold'**),  text=**"+"**, bg=**"yellow"**, command=lambda: btnClick(**"+"**)).grid(row=2, column=3) *#* btn4=Button(f2,padx=16, pady=16, bd=8, fg=**"black"**, font=(**'arial'**,20,**'bold'**),  text=**"4"**, bg=**"yellow"**, command=lambda: btnClick(4)).grid(row=3, column=0) btn5=Button(f2,padx=16, pady=16, bd=8, fg=**"black"**, font=(**'arial'**,20,**'bold'**),  text=**"5"**, bg=**"yellow"**, command=lambda: btnClick(5)).grid(row=3, column=1) btn6=Button(f2,padx=16, pady=16, bd=8, fg=**"black"**, font=(**'arial'**,20,**'bold'**),  text=**"6"**, bg=**"yellow"**, command=lambda: btnClick(6)).grid(row=3, column=2) *#* Subtraction=Button(f2,padx=16, pady=16, bd=8, fg=**"black"**, font=(**'arial'**,20,**'bold'**),  text=**"-"**, bg=**"yellow"**, command=lambda: btnClick(**"-"**)).grid(row=3, column=3) *#* btn1=Button(f2,padx=16, pady=16, bd=8, fg=**"black"**, font=(**'arial'**,20,**'bold'**),  text=**"1"**, bg=**"yellow"**, command=lambda: btnClick(4)).grid(row=4, column=0) btn2=Button(f2,padx=16, pady=16, bd=8, fg=**"black"**, font=(**'arial'**,20,**'bold'**),  text=**"2"**, bg=**"yellow"**, command=lambda: btnClick(5)).grid(row=4, column=1) btn3=Button(f2,padx=16, pady=16, bd=8, fg=**"black"**, font=(**'arial'**,20,**'bold'**),  text=**"3"**, bg=**"yellow"**, command=lambda: btnClick(6)).grid(row=4, column=2) *#* Multiply=Button(f2,padx=16, pady=16, bd=8, fg=**"black"**, font=(**'arial'**,20,**'bold'**),  text=**"\*"**, bg=**"yellow"**, command=lambda: btnClick(**"\*"**)).grid(row=4, column=3) *#* btn0=Button(f2,padx=16, pady=16, bd=8, fg=**"black"**, font=(**'arial'**,20,**'bold'**),  text=**"0"**, bg=**"yellow"**, command=lambda: btnClick(0)).grid(row=5, column=0) btnClear=Button(f2,padx=16, pady=16, bd=8, fg=**"black"**, font=(**'arial'**,20,**'bold'**),  text=**"C"**, bg=**"yellow"**, command=btnClearDisply).grid(row=5, column=1) *#no need for lambda* btnEquals=Button(f2,padx=16, pady=16, bd=8, fg=**"black"**, font=(**'arial'**,20,**'bold'**),  text=**"="**, bg=**"yellow"**, command=btnEqualsInput).grid(row=5, column=2) *#* Division=Button(f2,padx=16, pady=16, bd=8, fg=**"black"**, font=(**'arial'**,20,**'bold'**),  text=**"/"**, bg=**"yellow"**, command=lambda: btnClick(**"/"**)).grid(row=5, column=3) *#* root.mainloop() *#end* | | | | | |

*End of part one, two and three. Course will be continued next class. Until then work on exercises and read the reading materials provided.*

# References

(Under development)

# Appendix

## Printing patterns

Before thinking about patterns, let us recap about loop functions: The loop function in general helps to make your programming efficient. Instead of writing several lines for a repeating output, we can use the loop functions and have the same output with only fewer lines of codes. See the following examples:

Table , Pattern

|  |  |  |
| --- | --- | --- |
| The following two codes have the same outputs in the console. | | Printed output in the console: |
| print (**'^^^^^'**) print (**'^^^^^'**) print (**'^^^^^'**) print (**'^^^^^'**) print (**'^^^^^'**) print (**'^^^^^'**) print (**'^^^^^'**) print (**'^^^^^'**) print (**'^^^^^'**) print (**'^^^^^'**) | for i in range (1,11):  i += 1  print (**'^^^^'**) | ^^^^  ^^^^  ^^^^  ^^^^  ^^^^  ^^^^  ^^^^  ^^^^  ^^^^  ^^^^ |
| Class exercise: can you think of any other short lines of codes to do the same task?  Answer(example):  for i in range (1,11):  for x in range (5):  print (**'^'**, end=**''**)  print() |
| print(1\***'^'**) print(2\***'^'**) print(3\***'^'**) print(4\***'^'**) print(5\***'^'**) print(6\***'^'**) print(7\***'^'**) print(8\***'^'**) print(9\***'^'**) print(10\***'^'**) | x=0 i=**'^'** while x<11:  x += 1  print (x\*i) | ^  ^^  ^^^  ^^^^  ^^^^^  ^^^^^^  ^^^^^^^  ^^^^^^^^  ^^^^^^^^^  ^^^^^^^^^^ |
| Class exercise: can you think of any other short lines of codes to do the same task?  Answer (example):  for i in range (1,11):  for x in range (i):  print (**'^'**, end=**''**)  print() |

Printing patterns in python allows the programmer to practice visualization, logic, etc. in light of the fact, students should practice printing complex patterns in the console: