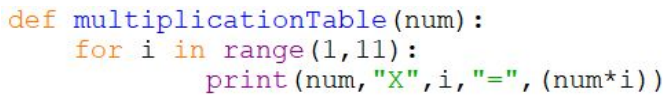
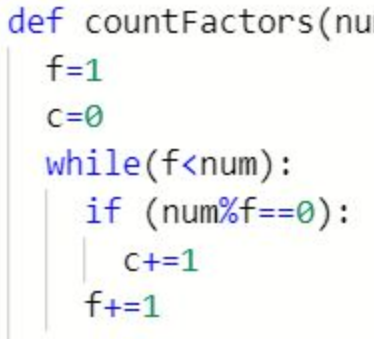
Session Code	CODR-912-BC-006
Module	Basic
Teaching Unit	Functions
Learning Outcome	Functions: in-built, custom.
Resources	Teacher: 1. Laptop along with audio and video exchange 2. Notebook and Pen(To note any development from session) Student Resources 1. Laptop along with audio and video exchange 2. Notebook and Pen(To keep note of important parts in the session)
Duration	50 Mins

Structure	**Warm-up** **Pace-up Activity** **Knowledge Transfer** **Student Led Activity** **Short Quiz** **Heads up tip for next class**	2 Mins 5 Mins 10 Mins 20 Mins 8 Mins 5 Mins

Step	Say	Perform
Warm up (2 Mins)	Hi *student name*, how are you? Are you excited for the class? Do you remember the last class?	Try to make the student speak. Engage with the student in conversation.
Interaction (5 Mins)	In the last class we learned about while and for loops. We also learned about range function.	Revise the concepts learned in the previous class.
	We have come across this word “function” a few times in the previous classes. Q. Do you wonder what it is? Q. Can you recall what other functions we came across? A. print(),input(),int(),len() or the latest we used range(). In all these cases, we gave some value to these functions and got some output. Let’s dig deeper and understand what a function is.	Let the student answer it.
Teacher shares the screen and open Repl.it		
Knowledge Transfer (10 min)	Function is a **block of code**, which runs when it is **called**, we can **pass values** into it, and it can **return values** or do something**.**	Copy paste the def and share in chat.
	Don’t worry if this sounds confusing. Let me explain it in detail. So, while writing many lines of code, there is always a chance that we need to code something which we have already coded. To reuse that part of code, we can convert it into a function and use it again just by writing the name of the function instead of all those lines of code.	

	Calling a function means writing a function name to run it.	
	Let's take the example of **range** function. We write the word range and then in brackets, we give values and in return we get some value. The values we give are called parameters or arguments of a function.	Explain giving real life analogy, eg: mixer grinder
	The functions we just mentioned like print(), len(), range() etc are all inbuilt functions.	
	We can even make functions of our own. Let's see the syntax of creating a function. **def** is the keyword used in defining a function. The function name follows similar rules as of naming a variable. We can't use spaces or special characters in it. Also keywords can't be used as function names. Arguments are the value we give to a function. **return** is a keyword which gives the output of the function, if any.	Teacher Activity 1: function syntax
	Now, let's create a function. In the last class we created a program to display the multiplication table of a number.	
	Today we will create a function to do the same thing, so that we can reuse the code, without writing it.	
	So, let's define the function. **def multiplicationTable(num):** Here multiplication Table is the function name.	
	num is an argument of this function. This is the variable which will store the value we pass into the function.	
	Below this line we write the program to display the multiplication table.	

		
	Now, to call the function, we simply write **multiplicationTable(12)** and it displays multiplication table of 12	
	Notice here we passed a value in the function and the function displayed some result, we didn't use the return function. Now it's your turn, you will create a function and use the return keyword in it.	
• **Stop sharing your screen** • **Ask the student to share screen and click on Student Activity 1:**		
	Here, you have to create a function, to count the no. of factors of a number.	Student Activity 1: function
	Do you remember how to find the factors of a number?? We run the loop from 1 to the given number and check if the number is divisible by it or not, if it is divisible we display the factor.	Let the student recall and answer, prompt the student with answer
	Now instead of displaying it as factor, we need to count the factors, for which we set another variable as a counter with initial value 0. Every time the condition for factor is true, we increase the value of the counter by 1. So at the end of the program, the counter will have the no. of factors of the given number.	
		

	But the function is not over yet, **return** keyword will return the value of the counter.	
	Now, to use this function, we can simply write the function name and pass a value **countFactors(8)** Nothing happens!! Can you guess why??	
	It's because the function has returned a value but we haven't done anything with it yet.	
	To see the value we can either store it in another variable or display it using the print function.	
	Now for the second activity, we need to first recall what Prime numbers are. Can you tell me what prime numbers are?? Number which is divisible by 1 and itself is called the prime number. So a prime number has only two factors	
	Here, we have already written a function to count factors, so can you tell me how we can check if it is prime or not.	Help the student write the program further.
• **Ask the student to close the repl.it window** • **Ask the student to click on student activity 2**		
	Help the student write the program, *It's actually a recall problem.* *This program needs a **for loop** which will pick each letter from the string **input** and **upper function** of string will convert it into upper case, for distinct letters, store each letter in a **set.***	Student Activity 1: Letter in Word
• **Ask the student to stop sharing the screen**		
Revision	Wow, today we used loops, conditionals, string functions and learned about functions.	
	Q. Can you tell me what the keyword **return** does? A. The keyword **return** is used to get a value from a function. Q. What rule do we follow while naming a	

	function? A. *all the rules of naming a variable.*	
Heads up for the next class	Now, that you are equipped with core concepts we will create interesting programs using different libraries in the next class.	
BID GOOD BYE & END CLASS		

Resources:

Activity	Name	Links
Teacher Activity 1	Repl link	https://repl.it/languages
Teacher Activity 2	Function syntax	https://drive.google.com/file/d/1Bt5UcBBKkGBhwwvyjBTV7fq0hgKosxoE/view?usp=sharing
Student Activity 1	function	https://repl.it/@ShailjaGupta/function
Student Activity 2	Letter in Word	https://repl.it/@ShailjaGupta/Letters-in-a-Word