**IT Career Switch Course – Software Development course:**

**Python:**

**Python is a general-purpose interpreted, interactive, object-oriented, high-level programming language.**

**Python is created by Guido van Rossum during 1985 – 1990.**

**Source code is available under the GNU General Public License (GPL).**

**It s named after a tv show called ‘Monty Python’s Flying Circus’ and not after Python the snake.**

**Python can be used on any platform, Windows, Mac, Linux, and will be using a Windows machine.**

**There are 2 popular versions out right now, 2 and 3 – we will use 3 in this course.**

**IDLE is the platform to create your python files inside.**

**IDLE:**

**IDLE is an integrated development environment for python, which has been bundled with the default implementation for a long time. IDE is a development environment which provides many features like coding, compiling, debugging, executing, autocomplete, libraries, in one place for the developer’s making tasks simpler. You can choose from a number of environments to develop your programs inside.**

**We will be using IDLE during this course along with the interpreter.**

**Take a look around:**

**Now that we have Python and our integrated enviropment is installed, lets take a look around and see what we have done so far.**

**Starting IDLE.**

**Look at the installation and how to start Python.**

**Straight from the interpreter.**

**Saving a file and have it opened from the editor (we can even look at a text editor, save and run).**

**Press F5 to run the code.**

**Lets create a working directory for this course.**

**Create our first program, Hello World.**

**First program:**

**Open python IDE**

**Type python in windows search and select Python 3.12**

**The interpreter should have opened**

**Type the following:**

**Print(“Hello World”)**

**Hello World! should have been displayed on the screen.**

**Lets change it around some and use IDLE to create our files, then we will execute the files with the interpreter. Hopefully you are allowed python to be added to the PATH environment variable during installation.**

**1: Open up IDLE.**

**2: Select file > New File**

**This opens the editor (where we will create our files)**

**3: In the editor type, print(Hello there”)**

**4: file, save as… and save your file in the location to be used for the course.**

**5: From the editor,click run > run module (or you can simply press F5)**

**Link from python.org ->** [**https://wiki.python.org/moin/PythonEditors**](https://wiki.python.org/moin/PythonEditors)

**Module 1 quiz:**

**Q1:**

**Python can only be used on a Linux system.**

**False**

**Q2:**

**Which of following is the latest version of Python?**

**3**

**The latest major version of Python is 3. However, Python does not have a version 4 as of now. Python 3.x is the ongoing version series with regular updates. The most recent release can be found within the Python 3.x series.**

**Top of Form**

**Bottom of Form**

**Q3:**

**Which of the following is an integrated development environment for Python?**

**IDLE**

**The correct answer is IDLE.**

**IDLE (Integrated Development and Learning Environment) is an integrated development environment for Python that comes bundled with Python itself.**

**Module 2:**

**Working with primitive data types:**

**In programming languages, a data structure is a way of organising and structuring pieces of data, and a data type is just a piece of data. Data structures can be divided into two categories.**

**Primitive and non-primitive data structures.**

**Primitive data structures are the simplest forms of representing data, where the non-primitive structures to organise and manage sets of primitive data.**

**We use the data types in our variables.**

**Let’s look at using variables on the next slide.**

**Variables:**

**Few rules follow when creating our variables.**

**Names can only contain letters, numbers, and underscores. This can only start with a letter or underscore, not a number. Spaces are not allowed in variables. Don’t use python keywords or function names as variables. Make them short, descriptive (nottoo short like a single letter).**

**Working with primitive data types;**

**Python has four primitive data types:**

**Integer**

**Float**

**String**

**Boolean**

**Print(type(DATA))**

**X = 10.32**

**Print(type(X))**

**<class ‘float’>**

**Integer:**

**Just as in mathematics an integer is a whole number that could hold a zero, positive, or negative value.**

**10, -221,0**

**Positive number.**

**X=2**

**Negative number**

**X=-2**

**Ero**

**X=0**

**Zero is neither positive nor negative.**

**Float:**

**A floating-point number or a float is a real number, meaning that it can be either a rational or an irrational number. Because of this, floating-point numbers can be numbers that can contain a fractional part, such as, 6.1 or 211.2871.**

**For the purposes of thinking of a float in a python program, it is a number, which contains a decimal point. To check if something is a float we can use the isinstance() function.**

**Isinstance(4, 5, float)**

**#print True**

**Float:**

**As a general rule integers don’t have a decimal point, whereas floats do, so 22 and 22.0 have the same value but they are different types. Using the float() function each string or integer could be changed to a float.**

**Number=10**

**Print(float(Number))**

**33.0**

**String:**

**A string is a sequence of one or more characters that can be either a constant or a variable.**

**Strings exist within either single quotes ‘ or double quotes “ in python, so to create a string, enclose a sequence of characters in quotes.**

**‘string in single quotes’**

**In python, strings are immutable so once it’s declared the value can’t be changed, instead a new object is created.**

**String:**

**first\_string = “Hello”**

**second\_string = first\_string**

**print(irst\_string)**

**print(second\_string)**

**# lets change first\_string**

**First\_string = “I have changed”**

**Print(second\_string)**

**Single quote example.**

**More on strings:**

**We can change the case in a string with a few methods.**

**Title()**

**Upper()**

**Lower()**

**Before we jump into that; A method is an action that pythoncan perform on a piece of data. Every method is followed by a set of parentheses.**

**Back to the code;:**

**Name = “Dhanyaal ashid”**

**Print(name.title())**

**Title():**

**This displays each word in the title case, where each word begins with a capital letter.**

**Upper()**

**This displays each word in all uppercase.**

**Lower():**

**This displays each word in all lowercase.**

**first\_name = “Dhanyaal”**

**print(first\_name.title())**

**print(first\_name.upper())**

**first\_name = “DHANYAAL”**

**print(first\_name.lower())**

**putting strings together:**

**concatenation:**

**fname = “Dhanyaal”**

**lname = “Rashid”**

**full\_name = fname + “” + lname**

**print(full\_name.title())**

**more on strings:**

**fname = “dhanyaal”**

**age = 30**

**print(fname.title() + “is” + str(age) + “years old”)**

**we cannot always account for extra spaces, especially in user supplied data python has a few built in methods for us to get rid of the extra spaces.**

* **Strip**
* **Istrip**
* **rstrip**

**strip:**

**This removes the white spaces from oth ends.**

**Lstrip:**

**This removes the extra white space from the right.**

**Game =’ baseball ‘**

**print(game.strip()**

**print(game.rstrip()**

**print(game.isstrip()**

**Boolean:**

**The Boolean data type can be one or two values, either true or false. Booleans are used to represent the truth values that are associated with the logic branch of mathematics, which forms algorithms in computer science. Whenever you see the data type Boolean, it will start with a capitalized B because it is named for the mathematician Gorge Boole.**

**The values True and False will also always be with capital T and F respectively, as they are special values in python.**

**The\_Boolean = 10>20**

**print(the\_Boolean)**

**#prints false**

**Comments:**

**This was not a great place to put this little tidbit, so I decided here was good as anywhere else. We need to talk about comments in your code. Comments are very important for teamwork and reminding yourself what you did months ago on a snippet of code. For python, the Mark (Round sign, #) indicates a comment.**

**#Program by dhanyaal rashid**

**# Creating a variable called name and assigning it the value of dhanyaal, then printing.**

**Name=”Dhanyaal”**

**Print(name)**

**Practice time:**

**1: Create a variable and assign it the integer value of 15. Then have the code display the type of varabled used.**

**2: create a variable and assign it a Boolean value so when you print the variable, you gt a value of true returned.**

**3: create a variable and assign it the float value of 6.36272. then have the code display the type of variable used.**

**4: create 3 variables all strings and concatenate them together.**

**5: modify lab 4 with comments at the top.**

**Quiz:**

**1:**

**Which is not a valid Python variable?**

**9Times**

**The correct answer is 9Times.**

**In Python, variable names cannot start with a digit. The other options are valid because they either start with a letter or an underscore.**

**2:**

**This is a whole number that could hold a zero, positive or negative value.**

**Integer**

**An integer is a whole number that can hold a zero, positive, or negative value.**

**3:**

**This is a sequence of one or more characters that can be either a constant or a variable.**

**String**

**String is a sequence of one or more characters that can be either a constant or a variable.**

**Module 3:**

**Working with multiple assignment statements:**

**Multiple assignment allows you to assign multiple variables at the same time in one line of code.**

**A, b=5, 10**

**Or**

**(a,b) = (5,10)**

**Here “a” is assigned the value of 5 and “b” is assigned the value of 10**

**A, b=5, 10.**

**Also, in python, you can assign one single value to several variables at the same time.**

**This allows ou to initialize several variales at once, which you can reassign later in the program yourself, or through user input.**

**A=b=c=3**

**Here all 3 variables are assigned the value of 3.**

**You’ll commonly see multiple assignment in for loops.**

**Numbers = (“First Num”: ‘1’,”Second Num”: ‘2’)**

**For key, value in numbers.items():**

**Print(F”key(key) has a value of (value)”)**

**Using an F-string or formatted string literal here, notice the f in the print statement.**

**Name = “Dave”**

**Age=”30”**

**Print(F”His name is (name) and he is (age) years old”)**

**Multiple assignments are actually fairly strict when it comes to unpacking the iterable we give to it if we ttry to unpack a larger iterable into a smaller number of variables, we’ll get an error**

**A,b=1, 2,3.**

**This should generate an error and this strictness is a good thing for debugging and troubleshooting.**

**If we’re working with an item that has a different size than we expected, the multiple assignment will fail loudly and we’ll hopefully now know about a bug in our program that we wereb’t yet aware of.**

**1:**

**X,y,z=10,20,30**

**2:**

**A,b,c,d=(33, “car”, 2.158, “hey”)**

**Print(a)**

**Print(b)**

**Print©**

**Print(d)**

**QUIZ:**

**1:**

**Python supports multiple assignment statements.**

**True**

**True. Python supports multiple assignment statements, allowing you to assign values to multiple variables in a single line. For example:**

**a, b, c = 1, 2, 3**

**2:**

**a = b = c = 3 ; What is the value of C in the equation?**

**3**

**The value of c in the equation a = b = c = 3 is 3. All three variables (a, b, and c) are assigned the value 3.**

**Module 4:**

**Convert types in python:**

**In python, there are two number data types; integers and floats.**

**Sometimes you are working on someone else’s code and will need toconvert an integer to a float or a vice versa, or you may find that you have been using an integer when wht you really need is a float. Python has built-in methods to allow you to easily convert integers to floats and floats to integers.**

**Int()**

**Float()**

**Pytons method float() will convert integers to floats. To use this function, add an integer inside of the paranthesis;**

**Float(22);**

**X=10**

**Print(float(x))**

**Python also has built in functions to convert floats to integers: int()**

**The int() function works similarly to the float() function. Add a floating-point number inside of the parentheses to convert it to an inteer.**

**Int(28, 17)**

**X=89.44**

**Print(int(x))**

**When converting floats to integers with the int() function, python cuts of the decimal and remaining numbers of a float to create an integer. We may round 390, python will not do this through the int() function.**

**X=89.72**

**Print(int(x))**

**String conversion using str():**

**Strings are a common form of data in computer programs, and we may need to convert strings to numbers or numbers to strings fairly often, especially when we are taking in user-generated data. We can convert numbers to strings through using the str() method. Pass either a number or a variable into the parenthesis of the method and then that numeric value will be converted to a string value.**

**Str(16)**

**Quiz:**

**Q1:**

**In Python, there are two number data types, which two are those data types?**

**Integers**

**Floats**

**In Python, the two number data types are:**

* **Integers: These represent whole numbers, which can be positive, negative, or zero (e.g., -1, 0, 42).**
* **Floats: These represent floating-point numbers, which are numbers that have a decimal point (e.g., 3.14, -2.0, 0.001).**

**Q2:**

**What would be the result of "float(16)" ?**

**16.0**

**The result of float(16) in Python would be 16.0.**

**Q3:**

**What would be the result of: "x = 16.89 print(int(x))"**

**16**

**Module 5:**

**A list is a data structure in python that is a changeable ordered sequence of elements. Each element or value tht is inside of a list is called an item. Lists are defined by having values between square brackets[].**

**Lists are great to use when you want to work with many related values. They enable you to keep data together that belongs together, condense your code, and perform the same methods and operations on multiple values at once.**

**Employees = [‘Sara’, ‘Tammy’, ‘Debbie’,‘John’, ‘Carrie’]**

**Each item in a list corresponds to an index number, which is an integer value, starting with the index number 0. Because each item in a python list has a correspondeing index number, we’re able to access and manipulate lists in the same way we can with other sequential data types.**

**Print(employees[3])**

**Quiz:**

**Q1:**

**This is a data structure in Python that is a changeable ordered sequence of elements.**

**List**

**Q2:**

**Lists are defined by having values between what?**

**[ ]**

**Lists in Python are defined by having values between [ ].**

**Module 6:**

**We can use indexing to change items within the list, by setting an index number equal to a different value. Tis gives us greater control over lists as we are able to modify and update the items that they contain.**

**Employees=[‘Sara’, Tammy’, ‘Debbie’, ‘John’, ‘Carrie’]**

**Employees[0]=’Mark’**

**Print(employees)**

**[‘Mark’, ‘Tammy’, ‘Debbie’, ‘John’, ‘Carrie’]**

**Operators can be used to make modifications to lists.**

**What about adding an item somewhere in the middle?**

**Items can be removed from lists by using the del statement. This will delete the value at the value at the index number you specify within a list.**

**You can loop through the list items using a for loop.**

**Quiz:**

**Q1:**

**This allows us to change items within a list**

**Indexing**

**Indexing allows us to change items within a list in Python.**

**Q2:**

**Items can be removed from lists by using this statement.**

**del**

**Module 7:**

**Sorting & reversing lists:**

**The sort() method sorts the list in ascending order by default.**

**Reverse that alphabetically.**

**Colors=[‘Blue’, ‘Red’. ‘Yellow’, ‘Green’]**

**Colors.sort()**

**Reverse that alphabetically.**

**Colors=[‘Blue’, ‘Red’. ‘Yellow’, ‘Green’]**

**Colors.sort(reerse=True)**

**Module 7:**

**When you use sort() it permanently changes the list, however you might not want the actual data to change, but just want you to work with.**

**To maintain the original order of a list, but present it in a sorted order we will use sorted()**

**Quiz:**

**Q1:**

**This method sorts the list ascending by default**

**sort()**

**This method sorts the list ascending by default: sort().**

**Q2:**

**When you use sort() it permanently changes the list.**

**True**

**True. When you use sort() on a list in Python, it permanently changes the order of the elements in that list.**

**Module 8:**

**Slicing lists:**

**Python has an amazing feature for that called slicing. Slicing can not only be used for lists or tuples, but custom data structures as well, with the slice object.**

**Nums=[1, 2, 3, 4, 5, 6]**

**Nums[2,5]**

**The 2 means to sort at second item in the list(note that the slicing index starts at 0).**

**The 5 means to end at the sixth item in the list, but not include it.**

**The colon in the middle is how python’s lists recognize that we want to use slicing to get objects in the list.**

**You can use a slice in a for loop if you want to loop through a subset of the elements in a list. In the next example, we loop through the first three players and print their names as part of a simple roster.**

**There is also an optional second clause that we can add that allows us to set how the lists index will increment between indexes that we’ve set.**

**That last colon tells python that we’d like to choose our slicing increment.**

**Quiz:**

**Q1:**

**If you saw nums[0:20:2] after seeing some numbers in a list, what is taking place?**

**Slicing**

**Slicing. In Python, nums[0:20:2] is an example of list slicing. It extracts elements from index 0 to 20, stepping by 2, from the list nums.**

**Q2:**

**What would be displayed after running the following code? "nums = [1,2,3,4,5] print (nums[2:4])"**

**3,4**

**The output of the code nums = [1,2,3,4,5] and print(nums[2:4]) would be:**

**[3, 4]**

**Module 9:**

**Woking with operators:**

**Operators are used toperform operations on ariables and values.**

* **Arithmetic operators**
* **Assignment operators**
* **Comparison operators**
* **Logical operators**
* **Logical operators**
* **Identity operators**
* **Membership operators**
* **Bitwise operators**

**Arithmetic operators**

**These are used with numeric values to perform common mathematical operations.**

**Assignment operators:**

**Assignment operators are used to assign values to variables.**

**Comparison operators:**

**Comparison operators are used to compare two values.**

**Logical Operators:**

**Logical operators are used to combine conditional statements.**

**Identity operators:**

**Identity operators are used to compare the objects, not if they are equal, but if they are actually the same object, with the same memory location.**

**Quiz:**

**Q1:**

**These are used to perform operations on variables and values.**

**Operators**

**Operators are used to perform operations on variables and values.**

**Q2:**

**Which operator is used with numeric values to perform common mathematical operations?**

**Arithmetic**

**Arithmetic**

**Arithmetic operators are used with numeric values to perform common mathematical operations such as addition (+), subtraction (-), multiplication (\*), and division (/).**

**Q3:**

**Which of the following are Assignment Operators?**

**=**

**+=**

**\*=**

**/=**

**All**

**Q4:**

**Which of the following are Logical Operators?**

**and**

**or**

**not**

**The Logical Operators in Python are:**

1. **and - Returns True if both statements are true.**
2. **or - Returns True if at least one of the statements is true.**
3. **not - Reverses the result, returning False if the result is true.**

**Therefore, the correct logical operators from the options provided are:**

* **and**
* **or**
* **not**

**The word "but" is not a logical operator in Python.**

**Module 10:**

**Determining Operator Precedence:**

**Just like in mathematics, we need to keep in mind that operators will be evaluated in order of precedence, not from left to right or right to left.**

**A = 5 +3\*2 16 or 11**

**For math: PEMDAS**

**Parentheses, exponents, multiple, divide, add, or subtract.**

**Quiz:**

**Q1:**

**Python operators will be evaluated in reverse order of normal mathematics order of operations**

**False**

**In Python, operators are evaluated in the same order as in normal mathematics. This order of operations is often remembered by the acronym PEMDAS (Parentheses, Exponents, Multiplication and Division (from left to right), Addition and Subtraction (from left to right)).**

**Q2:**

**In Python, which would be evaluated first?**

**Parentheses**

**In Python, Parentheses would be evaluated first.**

**Module 11:**

**Working with if statements:**

**Decision making is required when we want to evaluate a code only if a certain condition is satisfied. Here comes “If” to save the day. With some friends (compound conditions coming up):**

**X=12**

**Y=3**

**If x>y:**

**Print(“X isgreater than Y”)**

**The if statement and elif-else is powerful, but it’s only appropriate to use when you just need one test to pass. As soon as python finds one test that passes, it skips the rest of the tests.**

**This behaviour is beneficial, because it’s efficient and allows you to test for one specific condition.**

**Sometimes it’s important to check all of the conditions of interest though.**

**In this case, you should use a series of simple if statements with no elif or else blocks.**

**This technique makes sense when more than one condition could be True, and you want to act on every condition that is True.**

**Quiz:**

**Q1:**

**If we need to make a decision during the script running, we can use this statement.**

**If**

**If we need to make a decision during the script running, we can use the If statement.**

**Q2:**

**What is the result of the following code?  
x = 12  
y = 3  
if x > y:  
print("X is greater than Y")**

**X is greater than Y**

**The correct output would be:**

**X is greater than Y**

**Module 12:**

**Working with for loops:**

**Using for loops in computer programming allows us to automate and repeat similar tasks multiple times. A for loop implements the repeated execution of code based on a loop counter r loop variable. This means that for loops are used most often when the number of iterations is known before entering the loop, unlike while loops which are conditionally based.**

**Colors=[‘blue’, ‘red’, ‘pink’, ‘green’]**

**For x in colors:**

**Print(x)**

**This for loop sets up “x” as it’s iterating variable.**

**In loops, range() is used to control how many times the loop will be repeated. When working with range(), you can pass between 1 and 3 integer arguments to it:**

**Range(start, stop, step)**

**Start states the integer value at which the sequence begins, if this is not included then start begins at 0.**

**Stop is always required and is the integer that is counted up to but not included then start begins at 0.**

**Stop is always required and is the integer that is counted up to but not included.**

**Step sets how much to increase for decrease (or decrease in the case of negative number) the next interaction, if this is omitted then step defaults to 1.**

**Quiz:**

**Q1:**

**Which of the following allow us to automate and repeat similar tasks multiple times?**

**loops**

**The correct answer is loops.**

**Loops in programming allow you to automate and repeat similar tasks multiple times.**

**Q2:**

**What is the result of the following code?  
colors = ['blue', 'red', 'pink']  
for x in colors:  
print(x)**

**blue red pink**

**The correct answer is:**

**blue red pink**

**The code loops through the colors list and prints each color in the order they appear in the list: "blue", "red", and "pink".**

**Module 13:**

**While loop:**

**With the while loops we can execute a set of statements as long as a condition is true.**

**Quiz:**

**Q1:**

**Which type of loop can execute a set of statements as long as a condition is true?**

**While**

**The correct answer is:**

**While**

**2:**

**What is the result of the following code? a = 1  
while a < 4:  
 print(a)  
 a += 1**

**1, 2, 3**

**The correct answer is:**

**1, 2, 3**

**The code will print the values of a starting from 1 and incrementing by 1 each time until a is no longer less than 4.**

**Module 14:**

**Nesting for loops:**

**A nested for loop is a loop inside a loop.**

**The “Inner loop” will be executed one time for each iteration of the “Outer loop”.**

**1:**

**What is a loop inside a loop called?**

**Nested Loop**

**A loop inside a loop is called a Nested Loop.**

**2:**

**How many times will the inner loop be executed for each iteration of the outer loop in a nested loop?**

**Once**

**The inner loop will be executed once for each iteration of the outer loop multiplied by the number of iterations the inner loop runs. This means if the outer loop runs n times and the inner loop runs m times, the inner loop will execute m times for each of the n iterations of the outer loop, resulting in a total of n \* m executions.**

**Module 15:**

**Reading Files:**

**To read or write to a file, you need to open it first.**

**To open a file in python, use it’s built-in open() function.**

**This function returns a file object i.e. a handle.**

**You can use it to read or modify the file.**

**The mode argument is optional; ‘r’ will be assumed if it’s omitted.**

**Open() Function:**

**Remember to open the file, use the built-in open() function.**

**The open() function returns a file object, which has a read() method for reading the content of the file.**

**Writing files:**

**To write to an existing file, you must add a parameter to the open()) function:**

**“a” – append will append to the end of the file.**

**“w” – write this will overwrite any existing content.**

**Quiz:**

**Q1:**

**Which built-in function is used to ope a file in Python?**

**sort()**

**open()**

**file()**

**exec()**

**The built-in function used to open a file in Python is open().**

**Q2:**

**In the following code, what does the "r" represent for Python? a = open("demo.txt", "r")**

**access\_mode**

**Q3:**

**What would the following code do once executed? y = open("File.txt", "x")**

**Create the file File.txt**

**The code y = open("File.txt", "x") will create the file File.txt.**

**If the file already exists, it will error out. The "x" mode in Python is used for creating a new file; if the file already exists, an error will be raised.**

**Module 16:**

**Copying files:**

**Python provides in-built functions for easily copying files using the Operating System Shell utilities.**

**Following command is used to copy files.**

**Shutil.copy(src.dst)**

**Following command is used to copy files with MetaData information (MAC ties, permissions).**

**Shutil.copystat(src,dst)**

**We will have to import the shutil module, but what is a module?**

**Consider a module to be the same as a code library.**

**A file containing a set of functions you want to include in your application.**

**You can create your own or use the built-in modules with python.**

**Built-in is what we are going to use for now.**

**Quiz:**

**Q1:**

**In Python, what is A file containing a set of functions you want to include in your application referred to as?**

**Modules**

**In Python, a file containing a set of functions you want to include in your application is referred to as a module.**

**So, the correct answer is Modules.**

**Q2:**

**What is the following code meant to be used for? shutil.copy(src,dst)**

**Copy a file**

**The code shutil.copy(src, dst) is meant to be used to copy a file from the source path (src) to the destination path (dst).**

**So, the correct answer is Copy a file.**

**Q3:**

**What does the following code perform? Os.remove("demo.txt")**

**Delete demo.txt**

**The code os.remove("demo.txt") is used to delete the file named demo.txt.**

**So, the correct answer is Delete demo.txt.**

**Module 17:**

**Merging emails:**

**When we want to send the same invitations to many people, the body of the mail does not change.**

**Only the name (and maybe address) needs to be changed.**

**Mail merge is a process of doing this.**

**Instead of writing each mail separately, we have a template for body of the mail and a list of names that we merge together to form all mails.**

**Before we jump into an example, we need to touch on something again that will help us out and that is the with statement.**

**With statement:**

**With the “With” statement, you get better syntax and expectations handling.**

**In addition, it will automatically close the file. The with statement provides a way for ensuring that a clean-up is always used.**

**Alright, now that we have that under our belt some, Let’s jump into the main topic on this module, merging emails.**

**Quiz:**

**Python allows you to perform email merging.**

**True**

**True.**

**Python allows you to perform email merging using libraries such as smtplib for sending emails and email for constructing email messages. You can also use templates and tools like Jinja2 to automate and personalize email content for different recipients.**

**Q2:**

**Which statement can use to to open files where we get better syntax and exceptions handling, additionally it will close the file automatically?**

**with**

**The statement that provides better syntax and exception handling, and automatically closes the file after the operations are completed, is with.**

**So, the correct answer is with.**

**Module 18:**

**Reading console inputs and formatting outputs:**

**To receive information through keyboard, python uses either the input() or raw\_input() functions.**

**Input() is python 3.0**

**Raw\_input() is python 2.0**

**These functions have an optional parameter, commonly known as prompt, which is a string that will be printed on the screen whenever the function is called. When one of the input() or raw\_input() functions is called, the program flow stops until the user enters the iput via the command line. To actually enter the data, the user needs to press the enter key after inputting their string.**

**Input with numbers:**

**The input() function, by default, will convert all the information it receives into a string. Numbers used to be explicitly handled as such since they come in as strings originally.**

**Quiz:**

**Q1:**

**Which function do we use in Python 3.0 to receive input from the keyboard?**

**input()**

**In Python 3.0, the function used to receive input from the keyboard is input().**

**So, the correct answer is input().**

**Q2:**

**When using the input() function, the input will be automatically coverted to what data type?**

**string**

**Module 19:**

**Reading command line arguments:**

**Command line arguments allow you to enable programs to act in a certain way, for example, to output additional information, or to read data from a specified source, and to interpret this data in a desired format.**

**We are going to take a look at one way of handling command line arguments with python and that is using the sys module and sys.argv.**

**Sys.argv is a list in python, which contains the command-line arguments assed to the script.**

**With the len(sys.argv) function you can count the number of arguments.**

**With the str(sys.argv) function you can see the actual arguments being passed.**

**Quiz:**

**Q1:**

**Which arguments allow you to enable programs to act in a certain way, for example to output additional information, or to read data from a specified source?**

**command line**

**The arguments that allow you to enable programs to act in a certain way, such as outputting additional information or reading data from a specified source, are called command line arguments.**

**So, the correct answer is command line.**

**Q2:**

**The following line of code will print the name of our file."import sys print("The name of our file is:",(sys.argv[0]))"**

**True**

**Module 20:**

**Defining functions:**

**A function is a block of code which only runs when it is called.**

**You can pass data, known as parameters,into a function**

**A function can return data as a result.**

**In python a function is defined using the del keyboard.**

**Def first\_\_function():**

**Print(“Hello World”)**

**Quiz:**

**Q1:**

**This is a block of code which only runs when it is called.**

**function**

**The correct answer is function.**

**Q2:**

**In Python a function is defined using which keyword?**

**def**

**Module 21:**

**In Python, a function is defined using the def keyword.**

**So, the correct answer is def.**

**Module 21:**

**Using default arguents:**

**Python has a different way of representing syntax and default values for function arguments.**

**Default values indicate that the function argument will take the value if no argument value is passed during the function call.**

**The default value is assigned by using arguments = operator**

**Keyword = value**

**Any number of arguments in a function can have a default value.**

**Once we have a default argument, all the arguments to it’s right must also have default values.**

**So remember, non-default arguments cannot follow default arguments.**

**Quiz:**

**Q1:**

**The default value is assigned by using which assignment operator?**

**=**

**The default value is assigned by using the = assignment operator.**

**So, the correct answer is =.**

**Q2:**

**non-default arguments cannot follow default arguments**

**True**

**Module 22:**

**Using keyword and positional arguments:**

**Python allows functions to be called keyword arguments. When the functions are called in this way, the other (position) of the arguments can be changed. We can ix ositional arguments with keyword arguments during a function call. Keyword arguments must follow positional arguments. Having a positional argument after keyword arguments will result into errors.**

**Quiz:**

**Q1:**

**We can mix positional arguments with keyword arguments during a function call**

**True**

**Q2:**

**Keyword arguments must follow positional arguments**

**True**

**Module 23:**

**Handling exceptions:**

**This type of error occurs whenever the syntax is correct but the python code results in an error Not a syntax error like forgetting to close a parentheses or leaving off a colon.**

**Lets try print(0/0) and see what happens.**

**Python details what type of exception error was encountered**

**ZeroDivisionError**

**We can handle these exceptions using try blocks.**

**The words “Try” and “Except” are python keywords and are used to catch exceptions. The code within the try clause will be executed statement by statement**

**If an exception occurs, the rest of the try block will be skipped and the except clause will be executed.**

**Quiz:**

**Q1:**

**What happens when the following code is executed? print(0/0)**

**ZeroDivisionError**

**Q2:**

**We can handle exceptions like ZeroDivisionError using what?**

**TRY blocks**

**We can handle exceptions like ZeroDivisionError using TRY blocks.**

**Module 24:**

**Math Module:**

**In python a number of mathematical operations can be performed by inputting the math module. All listed here** [**https://docs.python.org/3/library/math.html**](https://docs.python.org/3/library/math.html)

**We can use this module to find the square root, sline, tangent, and powers of numbers and on and on (see link above).**

**Python has a built in module that you can use to make random numbers. The random module has a set of methods.**

**Quiz:**

**Q1:**

**Which module would you import inorder for your code to perform powers of numbers or square roots?**

**math**

**To perform powers of numbers or square roots in Python, you would import the math module.**

**Q2:**

**With the random module imported, we can use which function to generate a random integer?**

**randint()**

**With the random module imported, you can use the randint() function to generate a random integer.**

**Module 25:**

**In python, date, time and datetime classes provides a number of function to deal with dates, times, and time intervals.**

**Date and datetime are objects in python, so when you manipulate them, you are actually manipulating objects.**

**Whenever you manipulate dates or time, you need to import datetime function.**

**We will go over these classes for the datetime module.**

**Date – Manipulate just date (mnth, day, year)**

**Datetime – combination of time and date (Month, day, year, hour, second, microsecond)**

**Quiz:**

**Q1:**

**Whenever you manipulate dates or time, you need to import which module?**

**datetime**

**q2:**

**Which function do we use from the os module, to return the current working directory?**

**getcwd()**

**To return the current working directory using the os module, you use the getcwd() function.**

**End of course quiz:**

**57 questions:**

**Q1:**

**Python can only be used on a Linux system.**

**False**

**Python can be used on multiple operating systems, including:**

* **Windows**
* **macOS**
* **Linux**
* **Others (e.g., Unix-based systems, Android, etc.)**

**Q2:**

**Which of following is the latest version of Python?**

**3**

**Q3:**

**Which of the following is an integrated development environment for Python?**

**IDLE**

**The correct answer is IDLE.**

**IDLE (Integrated Development and Learning Environment) is an IDE that comes bundled with Python. It allows you to write, edit, and run Python code in a user-friendly interface.**

**Q4:**

**Which is not a valid Python variable?**

**9Times**

**The variable 9Times is not a valid Python variable.**

**Q5:**

**This is a whole number that could hold a zero, positive or negative value.**

**Integer**

**The correct answer is Integer.**

**Q6:**

**This is a sequence of one or more characters that can be either a constant or a variable.**

**String**

**The correct answer is String.**

**Q7:**

**Python supports multiple assignment statements.**

**True**

**Q8:**

**a = b = c = 3 ; What is the value of C in the equation?**

**3**

**In the statement a = b = c = 3, the value of c is 3.**

**Q9:**

**In Python, there are two number data types, which two are those data types?**

**Integers**

**Floats**

**The two main number data types in Python are:**

1. **Integers - These represent whole numbers (e.g., -10, 0, 25).**
2. **Floats - These represent numbers with decimal points (e.g., 3.14, -2.0).**

**Q10:**

**What would be the result of "float(16)" ?**

**16.0**

**The result of float(16) would be 16.0.**

**The float() function converts an integer to a floating-point number, so 16 becomes 16.0.**

**Q11:**

**What would be the result of: "x = 16.89 print(int(x))"**

**16**

**Q12:**

**This is a data structure in Python that is a changeable ordered sequence of elements.**

**List**

**The correct answer is List.**

**Q13:**

**Lists are defined by having values between what?**

**[ ]**

**Lists in Python are defined by having values between [ ] (square brackets).**

**Q14:**

**This allows us to change items within a list**

**Indexing**

**The correct answer is Indexing.**

**Q15:**

**Items can be removed from lists by using this statement.**

**del**

**Items can be removed from lists using the del statement.**

**Q16:**

**This method sorts the list ascending by default**

**sort()**

**The correct method to sort a list in ascending order by default is sort().**

**Q17:**

**When you use sort() it permanently changes the list.**

**True**

**Q18:**

**If you saw nums[0:20:2] after seeing some numbers in a list, what is taking place?**

**Slicing**

**The correct term for nums[0:20:2] is Slicing.**

**Q19:**

**What would be displayed after running the following code? "nums = [1,2,3,4,5] print (nums[2:4])"**

**3,4**

**Q20:**

**These are used to perform operations on variables and values.**

**Operators**

**The correct term is Operators.**

**Q21:**

**Which operator is used with numeric values to perform common mathematical operations?**

**Arithmetic**

**The operator used with numeric values to perform common mathematical operations is Arithmetic.**

**Q22:**

**Which of the following are Assignment Operators?**

**=**

**+=**

**\*=**

**/=**

**The following are Assignment Operators:**

* **= (simple assignment)**
* **+= (addition assignment)**
* **\*= (multiplication assignment)**
* **/= (division assignment)**

**Q23:**

**Which of the following are Logical Operators?**

**and**

**or**

**not**

**The following are Logical Operators:**

* **and**
* **or**
* **not**

**Q24:**

**Python operators will be evaluated in reverse order of normal mathematics order of operations**

**False**

**Q25:**

**In Python, which would be evaluated first?**

**Parentheses**

**In Python, Parentheses are evaluated first.**

**Q26:**

**If we need to make a decision during the script running, we can use this statement.**

**If**

**To make a decision during script execution, you can use the if statement.**

**Q27:**

**What is the result of the following code?  
x = 12  
y = 3  
if x > y:  
print("X is greater than Y")**

**X is greater than Y**

**Q28:**

**Which of the following allow us to automate and repeat similar tasks multiple times?**

**loops**

**The correct answer is loops.**

**Q29:**

**What is the result of the following code? colors = ['blue', 'red', 'pink'] for x in colors: print(x)**

**blue red pink**

**Q30:**

**Which type of loop can execute a set of statements as long as a condition is true?**

**While**

**The correct type of loop that executes a set of statements as long as a condition is true is While.**

**Q31:**

**What is the result of the following code? a = 1  
while a < 4:  
 print(a)  
 a += 1**

**1, 1, 1**

**1,2,3**

**Q32:**

**What is a loop inside a loop called?**

**Nested Loop**

**A loop inside a loop is called a Nested Loop.**

**Q33:**

**How many times will the inner loop be executed for each iteration of the outer loop in a nested loop?**

**Once**

**Q34:**

**Which built-in function is used to open a file in Python?**

**open()**

**The built-in function used to open a file in Python is open().**

**Q35:**

**In the following code, what does the "r" represent for Python? a = open("demo.txt", "r")**

**read\_mode**

**In the code a = open("demo.txt", "r"), the "r" represents the read\_mode.**

**Q36:**

**What would the following code do once executed? y = open("File.txt", "x")**

**Create the file File.txt**

**The code y = open("File.txt", "x") would result in an Error.**

**Q37:**

**In Python, what is A file containing a set of functions you want to include in your application referred to as?**

**Modules**

**In Python, a file containing a set of functions you want to include in your application is referred to as a Module.**

**Q38:**

**What is the following code meant to be used for? shutil.copy(src,dst)**

**Copy a file**

**The code shutil.copy(src, dst) is meant to Copy a file.**

**Q39:**

**What does the following code perform? Os.remove("demo.txt")**

**Delete demo.txt**

**The code os.remove("demo.txt") performs the operation to Delete demo.txt.**

**Q40:**

**Python allows you to perform email merging.**

**True**

**Q41:**

**Which statement can use to to open files where we get better syntax and exceptions handling, additionally it will close the file automatically?**

**with**

**The correct statement is with.**

**Q42:**

**Which function do we use in Python 3.0 to receive input from the keyboard?**

**input()**

**In Python 3.0, the function used to receive input from the keyboard is input().**

**Q43:**

**When using the input() function, the input will be automatically coverted to what data type?**

**string**

**Q44:**

**Which arguments allow you to enable programs to act in a certain way, for example to output additional information, or to read data from a specified source?**

**command line**

**The correct answer is command line arguments.**

**Q45:**

**The following line of code will print the name of our file."import sys print("The name of our file is:",(sys.argv[0]))"**

**True**

**Q46:**

**This is a block of code which only runs when it is called.**

**function**

**The correct answer is function.**

**Q47:**

**In Python a function is defined using which keyword?**

**def**

**In Python, a function is defined using the def keyword.**

**Q48:**

**The default value is assigned by using which assignment operator?**

**=**

**The default value is assigned in Python using the = assignment operator.**

**Q49:**

**non-default arguments cannot follow default arguments**

**True**

**Q50:**

**We can mix positional arguments with keyword arguments during a function call**

**True**

**Q51:**

**Keyword arguments must follow positional arguments**

**True**

**Q52:**

**What happens when the following code is executed? print(0/0)**

**ZeroDivisionError**

**When the following code is executed:**

**print(0/0)**

**it will raise a ZeroDivisionError.**

**Q53:**

**We can handle exceptions like ZeroDivisionError using what?**

**TRY blocks**

**We can handle exceptions like ZeroDivisionError using TRY blocks.**

**Q54:**

**Which module would you import inorder for your code to perform powers of numbers or square roots?**

**Math**

**To perform powers of numbers or calculate square roots, you would import the math module.**

**Q55:**

**With the random module imported, we can use which function to generate a random integer?**

**randint()**

**With the random module imported, you can use the randint() function to generate a random integer.**

**Q56:**

**Whenever you manipulate dates or time, you need to import which module?**

**datetime**

**Whenever you manipulate dates or time, you typically import the** *datetime* **module.**

**Q57:**

**Which function do we use from the os module, to return the current working directory?**

**getcwd()**

**From the os module, you use the getcwd() function to return the current working directory.**