Sr. No.	Practical Definition	
1.	**First module**	
	1.	Install JDK, setup Java environment and write a program to print —CODING IS FUN, ENJOY IT!.
	2.	Write a program in Java to generate first n prime numbers.
	3.	Write a program to enter two numbers and perform mathematical operations on them.
	4.	Write a program that calculate percentage marks of the student if marks of 6 subjects are given.
	5.	Write a program in Java to find maximum of three numbers using conditional operator.
	6.	Write a program to accept a line and check how many consonants and vowels are there in line.
	7.	Write a program to count the number of words that start with capital letters.
	8.	Create a class which asks the user to enter a sentence, and it should display count of each vowel type in the sentence. The program should continue till user enters a word “quit”. Display the total count of each vowel for all sentences.
	9.	Write an interactive program to print a string entered in a pyramid form. For instance, the string “stream” has to be displayed as follows: S St Str Stre Strea Stream
	10.	Write an interactive program to print a diamond shape. For example, if user enters the number 3, the diamond will be as follows: * * * * * * * * *
	12.	Write a Java Program to find area of Geometric figures using method Overloading.
	13.	Write a program in Java to create a simple scientific calculator using Math Class.
	14.	Write a program in Java to sort the elements of list so that they are in ascending order.
	15.	Write a program in Java to multiply two matrixes.
2.	**Second module**	
	1.	Write a program to create a “distance” class with methods where distance is computed in terms of feet and inches, how to create objects of a class.
	2.	Modify the “distance” class by creating constructor for assigning values (feet and inches) to the distance object. Create another object and assign second object as reference variable to another object reference variable. Further create a third object which is a clone of the first object.
	3.	Write a program in Java in which a subclass constructor invokes the constructor of the super class and instantiate the values.
	4.	Write a program in Java to develop overloaded constructor. Also develop the copy constructor to create a new object with the state of the existing object.
	5.	Write a program to show the difference between public and private access specifiers. The program should also show that primitive data types are passed by value and objects are passed by reference and to learn use of final keyword
	6.	Write a program to show the use of static functions and to pass variable length arguments in a function.

	7.	Develop minimum 4 program based on variation in methods i.e., passing by value, passing by reference, returning values and returning objects from methods.
	8.	Write a program that implements two constructors in the class. We call the other constructor using 'this' pointer, from the default constructor of the class.
	9.	Write a program in Java to demonstrate single inheritance, multilevel inheritance and hierarchical inheritance.
	10.	Java Program to demonstrate the real scenario (e.g., bank) of Java Method Overriding where three classes are overriding the method of a parent class. Creating a parent class.
	11.	Write a program that implements simple example of Runtime Polymorphism with multilevel inheritance. (e.g., Animal or Shape)
	12.	Write a program to compute if one string is a rotation of another. For example, pit is rotation of tip as pit has same character as tip.
	13.	Describe abstract class called Shape which has three subclasses say Triangle, Rectangle, Circle. Define one method area() in the abstract class and override this area() in these three subclasses to calculate for specific object i.e. area() of Triangle subclass should calculate area of triangle etc. Same for Rectangle and Circle.
	14.	Write a program in Java to demonstrate multiple inheritance.
	15.	a) Write an application that illustrates method overriding in the same package and different packages. b) Also demonstrate accessibility rules in inside and outside packages.
3.	**Third module**	
	1.	Read a content from file: calculate number of sentences, words and characters from the file.
	2.	Read from a file convert it to uppercase and save it into another file.
	3.	Remove duplicate lines from a File.
	4.	Create a class called Student. Write a student manager program to manipulate the student information from files by using FileInputStream and FileOutputStream
	5.	Refine the student manager program to manipulate the student information from files by using the BufferedReader and BufferedWriter
	6.	Write a program to manipulate the information from files by using the Reader and Writer class. Assume suitable data.
	7.	Write a program "DivideByZero" that takes two numbers a and b as input, computes a/b, and invokes Arithmetic Exception to generate a message when the denominator is zero.
	8.	Write a program to show the use of nested try statements that emphasizes the sequence of checking for catch handler statements.
	9.	Write a program to create your own exception types to handle situation specific to your application (Hint: Define a subclass of Exception which itself is a subclass of Throwable).
	10.	Write a small application in Java to develop Banking Application in which user deposits the amount Rs 1000.00 and then start withdrawing of Rs 400.00, Rs 300.00 and it throws exception "Not Sufficient Fund" when user withdraws Rs. 500 thereafter.
	11.	Write a program to handle ArrayIndexOutOfBounds exception for binary search.
	12.	Write a Java Program that demonstrates thread class and few methods.
	13.	Write a program to demonstrate thread example by implementing runnable interface.
	14.	Write a program to demonstrate priorities among multiple threads.
	15.	Write a program to demonstrate multithread communication by implementing synchronization among threads (Hint: you can implement a simple producer and consumer problem).
4.	**Fourth module**	
	1.	Write a program to demonstrate different Window handling events.
	2.	Write a program to demonstrate different mouse handling events like mouseClicked(), mouseEntered(), mouseExited(), mousePressed, mouseReleased() and mouseDragged().

	3.	Write a program to demonstrate different keyboard handling events.
	4.	Write a program to generate a window without an applet window using main() function.
	5.	Write a program to demonstrate the use of push buttons.
	6.	WAP to create a Menu using the frame.
	7.	WAP to create a Frame that display the student information.
	8.	WAP to create a Dialogbox.
	9.	WAP to implement the FlowLayout and BorderLayout.
	10.	WAP to implement the GridLayout and CardLayout.
	11.	WAP to implement the GroupLayout and BoxLayout.
	12.	Write a program that demonstrates the life cycle of an applet.
	13.	WAP to demonstrate System clock.
	14.	WAP to demonstrate Painting in applet.
	15.	WAP to demonstrate Graphics in applet.