Three popular testing techniques:

1. Static Testing: Static testing is used during preliminary investigation, design and analysis phases. This approach involves testing various kinds of documents such as UML Diagrams, specifications etc

2. White box testing: The objective of white box testing is to verify the correctness of the software'sstatements, code paths, conditions, loops, and data fl ow

3. Black box testing: The objective of black box testing is to verify the correctness of the software's behavior that directly supports daily business activity. It involves testing output results, using various inputs. Does not deal with the source code

Testing Strategy

Testing strategy involves using an intelligent combination of above testing techniques

Artifact to be tested	**Objective of testing**	**Testing technique(s) used**		
		Static Testing	Whitebox Testing	Blackbox testing
UML Class diagram	1. Relationships between classes should be logical 2. Class's should successfully fulfill all their responsibilities 3. Class methods and attributes should be named logically, so that the class diagram can be easily interpreted by others (for example, the development team)	Static testing should be used to identify incomplete, incorrect or conflicting information in the diagrams For this assignment,*Desk Checking method* should be used to read, interpret & statically test the diagrams.		
UML Sequence Diagram	1.Should reflect runtime interactions between various components of the system. The diagram should also reflect interactions between actors and the system 2.The diagram must reflect order of interactions in a correct and logical way			
Class Name: Map	1. **readTowns(townFileName)** 1.a) To check if the function reads information from locations.txt file correctly 1.b) To check if function creates *Town objects* correctly, based on data read from locations.txt		*printData(String data)* function should be introduced, which prints each line read from the text file. This will help compare the output produced by program with the text file. This function should be called by *readTowns(townFileName)* and	Black box testing should be used to check if the desired output is produced by the functions

	2.**readRoads(roadFileName);** 2.a) To check if function reads information from roads.txt file correctly 2.b) To check if function creates *Road object* correctly, based on data read from roads.txt. 2.c) To check if *Road objects* created are properly assigned to the *Town* objects they connect		*readRoads(roadFileName); methods,* as they read each line from text file	
Class Name: Town	If class attributes (name, latitude, longitude, roads) have been initialized correctly		A function *printTown()* should be introduced in *Town class*, which prints details about each *Town object* created. This *printTown()* function should be called from *readTowns(String townFileName)* function, immediately after each *Town object* is created & initialized	
Class Name: Road			A function *printRoad()* should be introduced in *Road class*, which prints all the details about the road, which connects two towns. The function should also print names of the *Towns* that the *Road* connects. This function should be called from *readRoads(String fileName)* function, immediately after each *Road object* is created & initialized	
Class Name: Route	If route has been calculated properly or not. And to verify that there is no duplication of any leg		A function called *printRouteRoad()* should be introduced, which loops through each *Road object* (contained by *Route object)* and calls *printRoad()* function for each *Road object*	
Class Name: RouteCalc	If *Route objects* have been created and calculated properly or not		This class keeps a list of all *Route objects* created. So, a function called *printAllRoutes()* should be introduced, which loops	

			through each *Route object,* and calls *printRouteRoad()* method of each *Route object*	
Entire System	1. To check if the program can correctly calculate and display all possible routes between 2 towns, along with total distance of each route 2. To check if program correctly calculates shortest route			1. Various combinations of origin and destination towns should be selected by running the program 2. The results should be noted down and compared against the data provided in the assignment brief