Category	Total Marks	Remarks	Partial Marks
Correct Working	6	Should show correct output and never crash for all given test cases. Check that the players cannot vote themselves out and that mafia can't kill mafia and detective don't test on themselves. Little bit of variations in output is fine as long as its not incorrect. To check: (The output is given roundwise + The players are randomly alloted + The game ends properly with output of each players character + Check the game play and make sure it matches the game description)	**5*1=5**
		Random voting and simulation of game even if the user has been eliminated	**0.5*2=1**
Identifying Classes	3	These classes should exist in student solution: Character, Mafia, Detective, Healer, commoner, player	**0.5 x 6 = 3**
Inheritence Tree	3	Character class should be **abstract** as no one should be able to create objects of this class	**1**
		Mafia should be inherited from character	**0.5**
		Detective should inherit from character	**0.5**
		Healer should inherit from character	**0.5**
		Commoner should inherit from character	**0.5**
Inheritence Data Sharing, Object class	3	**Attribute HP, playerName should be in the character class**	**0.5**
		The attributes of base classes should be marked private (or protected)	**0.5**
		Use of InstanceOf for checking similarity of objects	**1**
		Use of toString while printing the player name	**1**
Inheritence Tree Methods and their implementation	4.5	Character class should have an abstract method "Strategy/Role"(Or any related method)	**0.5**
		Mafia should provide a concrete implementation of "Strategy/Role" where it takes an input from the player(If player is mafia) or chooses a random player(not a mafia) and decrease their HP	**0.5**
		Detective class should also provide concerete implementation of the above "Strategy/Role" method and take input from the player (if he is a detective or test a random player) or choose a random player(not detective) and if the test if positive, the mafia should be voted out.	**0.5**
		Healer class should also provide concerete implementation of the above "Strategy/Role" method and take inout from the player (if he is a detective or test a random player)	**0.5**

		If the player is killed, the game should not ask for inputs and simulate itself.	**0.5**
		If the player is commoner there should be no strategy. In such case the only input is taken during voting.	**0.5**
		All overridden methods in subclasses should be annotated with @Override	**1.5**
Miscellaneous	1.5	The game shouldn't terminate randomly and after the game ends the proper output should be there as defined in the assignment	**1.5**
Total	**21**		**21**