AP Midsem Set-3 Rubric (20)

Question	Criteria	Description	Marks
Q1 (8)	Professor and Student Classes	The 2 classes should be defined with private attributes, parametrised constructors and toString() methods.	2 (1 mark for each class)
	Course Class and Object Creation.	The 2 student objects should be created INSIDE the Course class (Composition). The Professor object should be created in main(), and passed to constructor of Course class as argument (Association). Also, the objects should be printed as a whole (overriding toString() method), and the attributes should NOT be printed separately. This means that there should be 3 print statements, not 6. If attributes are printed separately (correctly) and toString() is not defined, award 0.5/1.	3 (1+1+1), 1 for composition, 1 for association, 1 for printing properly.
	User Input	3 Strings and 3 integers should be taken as user input. dispStudents() and dispProf() methods of Course class should also be called in main.	1 (0.5+0.5)
	Correct Implementation	Code should give correct output for given test case. Binary marking to be followed. No marks to be awarded for partially correct outputs. The details of students should be printed in dispStudents() method, and details of professor should be printed in dispProf() method.	2
	Classes created	Give Zero marks if Generic class is not created.	1
	Encapsulation / correct use of modifiers	With proper reasons encapsulation should be used. (0.5 for using encapsulation and 0.5 for reasoning.)	1
		Use of Instanceof keyword	2
		There are at least 2 errors that have been handled.	2

Q2 (12)	Class methods	Check if each method is working properly or not. Don't give any marks if the methods don't run or the student is not able to run the program. Mark as follows: 1. setCoefficients + getCoefficients = 0.5 + 0.5 2. addPolynomial, subtractPolynomial = 0.5 + 0.5 3. getPolynomialGradient = 1 4. printPolynomial = 0.5 5. takeInput = 0.5 No new objects should be created at the time of demo to check for the functionality. Use the existing objects written by the student to check for correctness of functionality.	4 (binary marking to be followed if a method doesn't work)
	Correct output on test case	Check if the code is executing and giving proper results on the given test case. No marks to be given if it's not being executed.	2 (binary marking if output is not completely correct)
		Total	**20**