

# Java module 3

## Exercises Day 5

1 - Exceptions	Catching exceptions
Instructions	<p>You are the teacher of a newly formed class. To manage the students' information, create a Student class with attributes such as:</p> <ul style="list-style-type: none"><li>• name</li><li>• surname</li><li>• age</li><li>• grades (an array of 3 integers).</li></ul> <p>Additionally, add methods to modify these attributes.</p> <p>Create a class with the following code to test your implementation:</p> <pre>public class ExceptionTest {     public static void main(String[] args) {         Scanner scanner = new Scanner(System.in);          System.out.println("&gt;&gt;&gt; New Student &lt;&lt;&lt;");         System.out.print("Enter student name: ");         String name = scanner.nextLine();         System.out.print("Enter student surname: ");         String surname = scanner.nextLine();         System.out.print("Enter student age: ");         int age = scanner.nextInt();         scanner.nextLine();         //Create the student object         Student student = new Student(name, surname, age);         System.out.println("The student was created.");          //Initialize a new grades array         int[] grades = new int[3];         int index = 0; //Index to keep track of the         position in the grades array         String choice = "y";         while (choice.equals("y")) {             System.out.print("Enter a grade for the student: ");             int grade = scanner.nextInt();             scanner.nextLine();             //Add the grade to the array             grades[index] = grade;             index++;             System.out.print("Grade added. Would you like to add another grade (y/n)? ");             choice = scanner.nextLine();         }         //Once the user is done adding grades, update the student object         student.setGrades(grades);         System.out.println("Student grades were updated.");     } }</pre>

	<p>Make the necessary updates to solve gracefully the following cases:</p> <ul style="list-style-type: none"> <li>- If the user enters an age that is not a number.</li> <li>- If the user enters more than 3 grades.</li> </ul>
Expected output	<pre>&gt;&gt;&gt; New Student &lt;&lt;&lt; Enter student name: Carol Enter student surname: Green Enter student age: abc The age needs to be a number. Enter student age: 27 The student was created. Enter a grade for the student: 8 Grade added. Would you like to add another grade (y/n)? y Enter a grade for the student: 9 Grade added. Would you like to add another grade (y/n)? y Enter a grade for the student: 8 Grade added. Would you like to add another grade (y/n)? y Enter a grade for the student: 7 Not allowed to add more grades. Student grades were updated.</pre>

2 - Exceptions	Throwing exceptions
Instructions	<p>Run the program from the previous exercise and enter a negative number for the student's grade. It will allow it!</p> <p>Fix the program such that it isn't possible to enter negative grades. The student method setGrades should throw an exception when this happens.</p>
Expected output	<pre>&gt;&gt;&gt; New Student &lt;&lt;&lt; Enter student name: Carol Enter student surname: Green Enter student age: 27 The student was created. Enter a grade for the student: -8 A grade needs to be a number between 1 and 10. Enter a grade for the student: 8 Grade added. Would you like to add another grade (y/n)? n Student grades were updated.</pre>