

**TEAM PROGRAMMING CHALLENGE RUBRIC - "JUDGING CRITERIA" – ACTE**  
**Alabama Consortium for Technology in Education**

<b>TEAM PROGRAMMING CHALLENGE</b>	<b>MINIMAL</b>	<b>PARTIAL</b>	<b>MASTERY</b>	<b>RANK</b>
<b>Judge Discretionary Points *</b> 0 – 10	0 – 5	6 – 9	10	
<b>READABILITY*</b> 0 – 15 <b>Is the code arranged in an orderly manner?</b>	0 – 5 Code difficult to read with little to no structure.	6 – 10 Some or most of the code is readable. Code may be overly verbose or unclear considering expected student knowledge at this level.	11 – 15 All code readable throughout and not obfuscated considering expected level of student knowledge.	
<b>DESIGN CREATIVITY*</b> 0 – 20 <b>Did student(s) use a higher level of creativity throughout the design process and presentation?</b>	0 – 7 Minimal levels of creativity shown in the program design and oral presentation. No real design in mind for the program.	8 – 14 Displays lower level of creativity in the design process and oral presentation. Have basic outline of the program and can discuss how to implement their design.	15 – 20 Displays high level of creativity in entire design process and oral presentation. Unique, well planned and creative. Efficient solution identified for all edge cases.	
<b>EXECUTION/PERFORMANCE*</b> 0 – 25 <b>Did the program work for the intended purpose?</b>	0 – 9 Little to none of the program fit the purpose. The program has errors making the code practically unstable or will not even compile.	10 - 17 Elements of the project are not cohesive. Program does not fully serve its intended purpose. The program has some errors and/or has a brute force approaches and elegant solutions.	18 – 25 Outstanding solution. Easy to understand and maintain. Code comments help troubleshooting. Code organized. Program runs with no errors and is primarily composed of elegant solutions.	
<b>UNDERSTANDING*</b> 0 – 30 <b>Do students demonstrate a solid understanding of their chosen programming language and common software concepts? (Functions, recursion, different types of iteration)</b>	0 – 10 Displays little to no understanding of the program language. Code used overly simple concepts and code quality suffered. (i.e. Hard coding N iterations when a loop fits naturally)	11 – 20 Partially skilled and shows incomplete understanding of the program language. Code uses concepts like functions, switch statements, etc. Room for improvement.	21 – 30 Can explain analysis of alternative solutions to get most efficient solution. Mastery of programming language chosen and common software concepts.	
<b>COMMENTS</b>			<b>TOTAL SCORE</b>	

**TEAM PROGRAMMING CHALLENGE RUBRIC - "JUDGING CRITERIA" – ACTE**  
**Alabama Consortium for Technology in Education**

--	--	--	--	--