Bayview Glen School

85 Moatfield Drive
Toronto, ON

Name: _____
Date: Monday, October 25, 2010
Teacher: Mr. DesLauriers
Course: Introduction to Computer Science, Grade 11, University Preparation, Block 7

Selection and Iteration Summative Assignment

Students are to create a Blackjack game using the Java and the Eclipse IDE. The game will be completely text based and will involve the user playing against an automated dealer.

Overall Expectations

11A1	demonstrate the ability to use different data types, including one-dimensional arrays, in computer programs
11A2	demonstrate the ability to use control structures and simple algorithms in computer programs
11A3	demonstrate the ability to use subprograms within computer programs
11A4	use proper code maintenance techniques and conventions when creating computer programs
11B3	design algorithms according to specifications

Specific Expectations

11A1.1	use constants and variables, including integers, floating points, strings, and Boolean values, correctly in computer programs
11A1.3	use assignment statements correctly with both arithmetic and string expressions in computer programs
11A1.4	demonstrate the ability to use Boolean operators (e.g., AND, OR, NOT), comparison operators (i.e., equal to, not equal to, greater than, less than, greater than or equal to, less than or equal to), arithmetic operators (e.g., addition, subtraction, multiplication, division, exponentiation, parentheses), and order of operations correctly in computer programs
11A2.1	write programs that incorporate user input, processing, and screen output
11A2.2	use sequence, selection, and repetition control structures to create programming solutions
11A2.3	write algorithms with nested structures (e.g., to count elements in an array, calculate a total, find highest or lowest value, or perform a linear search)
11A3.1	demonstrate the ability to use existing sub-programs (e.g., random number generator, substring, absolute value) within computer programs
11A4.2	use workplace and professional conventions (e.g., naming, indenting, commenting) correctly to write programs and internal documentation
11A4.5	demonstrate the ability to validate a program using a full range of test cases
11B1.3	use the input-process-output model to solve problems
11B2.2	use appropriate vocabulary and mode of expression (i.e., written, oral, diagrammatic) to describe alternative program designs, and to explain the structure of a program
11B2.5	design user-friendly software interfaces (e.g., prompts, messages, screens, forms)
11B3.2	solve common problems (e.g., calculation of hypotenuse, determination of primes, calculation of area and circumference) by applying mathematical equations or formulas in an algorithm
11B4.5	use a variety of methods to debug programs (e.g., manual code tracing, extra code to output the state of variables)
11C2.1	use an operating system to organize computer programs and files logically on local and shared drives
11C3.1	demonstrate an understanding of an integrated software development environment and its main components (e.g., source code editor, compiler, debugger)
11C3.2	work independently, using support documentation (e.g., IDE Help, tutorials, websites, user manuals), to design and write functioning computer programs

Criteria	Level 1 (50% - 59%)	Level 2 (60% - 69%)	Level 3 (70% - 79%)	Level 4 (80% - 100%)	Mark

Knowledge and Understanding					
comprehend solution requirements	comprehends a few of the solution requirements	comprehends some of the solution requirements	comprehends many of the solution requirements	comprehends all or almost all of the solution requirements	
understands the correct data type to use in a particular situation	chooses the correct data type to represent data with limited success	chooses the correct data type to represent data with some success	chooses the correct data type to represent data with most of the time	chooses the correct data type to represent data all of the time	
Thinking					
validate a program using appropriate data	is able to validate a program using appropriate data with limited comprehension	is able to validate a program using appropriate data with adequate comprehension	is able to validate a program using appropriate data with proficient comprehension	is able to validate a program using appropriate data with superb comprehension	
solve simple problems using a programming language	is able to solve simple problems using a programming language with limited success	is able to solve simple problems using a programming language with some success	is able to solve simple problems using a programming language with considerable success	is able to solve simple problems using a programming language with excellent success	
Demonstrate use of repetition and iteration structures	rarely uses the correct type of repetion structure correctly.	uses the correct repetition structure but uses it incorrectly on occasions.	uses the correct repetition structure efficiently most of the time.	uses the correct repetition structure efficiently on a regular basis.	
Communication					
correctly document all programs	correctly documents a few areas of the program	correctly documents some areas of the program	correctly documents most areas of the program	correctly documents all or almost all areas of the program	
uses appropriate conventions when choosing identifier names	chooses appropriate names for identifier rarely	chooses appropriate names for identifier in some situations but not o a regular basis	chooses appropriate names for identifier most of the time	chooses appropriate names for identifier always	
uses correct programming covensions to make code easy to read	code is rarely indented and spaces correctly	code is indented and spaces correctly some of the time	code is indented and spaces correctly most of the time	code is indented and spaces correctly throughout the program	
Application					
demonstrate use of internal documentation and defined standards	demonstrates limited use of internal documentation and defined standards	demonstrates some use of internal documentation and defined standards	demonstrates considerable use of internal documentation and defined standards	demonstrates excellent use of internal documentation and defined standards	
use appropriate programming structures and conventions	rarely uses appropriate programming structures and conventions	sometimes uses appropriate programming structures and conventions	often uses appropriate programming structures and conventions	always or almost always uses appropriate programming structures and conventions	

write programs that use decision making structures	is able to write programs that use decision making structures with limited success	is able to write programs that use decision making structures with adequate success	is able to write programs that use decision making structures with considerable success	is able to write programs that use decision making structures with excellent success	
write programs that use various comparison structures	is able to write programs that use various comparison structures with limited success	is able to write programs that use various comparison structures with adequate success	is able to write programs that use various comparison structures with considerable success	is able to write programs that use various comparison structures with outstanding success	