

PROJECT CODING RUBRIC SCORE BREAK DOWN:

Checklist	Points Earned	Where in Code it is found
Compilable	1	Compiles with no errors
Runtime Errors	1	No runtime errors
Readability	1	Code is fairly easy to read
Math/Technical Knowledge from other areas	2	Use of counters to keep count of bets, multiplication to calculate payouts, percentage difference (economics) to see how much increase/decrease was earned
Independency	1	Compilable on any OSES using any IDEs or compilers
Code Organization	1	Functions and classes are separated into source and headers files
Robustness	1	Code captures edge-cases and wrong inputs
Resource Usages	1	No memory leaks/reasonable RAM and CPU consumption
Write, Compile, and Include Your Own Libraries	0	Did not compile a personal library (used an external 3 rd party library OpenSSL)
All Primitive Data Types	0.5	Example where each is used: <ul style="list-style-type: none">• Char• Int (unsigned)• Int (Signed)• Float• Double• String
Global and Static Variables	0.5	Global Variables (in main.cpp): <ul style="list-style-type: none">• bool imageDisplayed – line 5• std::string projectName – line6• int initialBalance – line7 Static Variables <ul style="list-style-type: none">• static int dealer_last_name – line 41
Bitwise Operators	0.5	Used bitwise XOR operator (^) <ul style="list-style-type: none">• Found in randomNumber.cpp in line 5
Conditional AND Iterative Statements	0.5	Use of Conditional (IF) and Iterative (WHILE) statements can both be found in main.cpp
Functions	0.5	Total of 9 functions <ul style="list-style-type: none">• BettingOptions()• EndOfGameExpression()• getBettingOptionParameters()• Image()• initializePlayerData()• main()• output_DealerName()• randomNumber()• writePlayerDataToFile()
Preprocessor	1	Used all of the following preprocessors: <ul style="list-style-type: none">• #include <iostream>• #include <cstdlib>• #include <ctime>• #include <vector>• #include <string>• #include <limits>• #include <tuple>• #include <fstream>• #include <regex>• #include <array>• #include <thread>• #include <mutex>
Pointers	1	Example of use of pointer: <ul style="list-style-type: none">• main function, line 52, playerPtr

References	1	<p>Example of use of references:</p> <ul style="list-style-type: none"> Two references used in InitializePlayerData.cpp function playerName and initialBalance are both passed by reference
Function Overloading	0.5	<p>output_DealerName function is overloaded</p> <ul style="list-style-type: none"> One version can take strings as input Other version can take integers as input
Lambda Function	1	<p>Example use of lambda function:</p> <ul style="list-style-type: none"> Main function, line 62, lambda function name NumBetsTotal
Template Function	1	<p>Example use of template function:</p> <ul style="list-style-type: none"> Main function, line 13, template function name AmountWon AndExpression
Containers	2	<p>Used three different types of containers:</p> <ul style="list-style-type: none"> Tuple Container in getBettingOptionParameters.cpp, line 4 Vector Container in main.cpp, line 97 Array Container in output_DealerName.cpp, line 5
Struct/Class	2	<p>Three classes are created:</p> <ul style="list-style-type: none"> Bet_Check – in Bet_Check.h Bet – in Bet.h Player – in Player.h
Objects	1	One object per class was created
Operator Overloading	2	(+) operator, (-) operator, and (/) operator were all overloaded in the Player class found in the Player.h file
Inheritance	1	One inheritance found in Bet_Check class inheriting from Bet class
Virtual Functions	2	Four virtual functions can be found in Bet.h header file/class
GUI	1	Use of a text based GUI, as well as OpenSDL
File System Handling	0.5	File system handling can be found in main function. Project reads from “player_data.txt” file and writes to the file as well to keep track of wins/losses by player
Regular Expression	1	Use of regular expression (win lose) found in EndOfGameExpression.cpp function, line 25 through 30
Concurrent Programming	1	<p>Two threads are used in the program</p> <ul style="list-style-type: none"> Main thread imageThread
TOTAL POINTS	30.5/35	Satisfies requirement of at least 30 points, obtains full credit for this section