This assignment will focus on top-down development using objects. Testing and debugging methods will have to be used and explained during the discussion portion of the assignment review. You must use multiple files to create this game: 1 file is the client code, 2 other files are for classes. The client code should be very concise – all the work should be done by the classes which are in the other 2 files.

You will create an application to play Rock, Paper, Scissors against the computer. The rules of the game are that Rock dulls Scissors (Rock wins), Scissors cut Paper (Scissors win) and Paper covers Rock (Paper wins).

**Specifications**

The game is played by the user selecting the number of rounds to play, to a maximum of 5. The user will start the game by throwing either rock, paper or scissors with their hand. The game randomly selects either rock, paper or scissors for the computer’s throw. Then the hands are compared and a winner selected based on the rules. If both throw the same object, you will be prompted to reselect and throw. You must track the number of wins for the player and the number of wins for the computer. At the end of the 5th round, the game will declare an overall winner. If an even number of rounds was initially selected and there is a tie in the number of wins, a tie-breaker match will be played to determine the overall winner.

**Sample Screen Layout**

Your screen must look like this or better

**Welcome to Rock, Paper, Scissors!**

**How many rounds would you like to play? 3**

**Enter your throw (1=Rock, 2=Paper, 3=Scissors): 3**

**You throw Scissors**

**The computer throws Paper**

**You WIN! Score – you: 1 Score – computer: 0**

**Enter your throw (1=Rock, 2=Paper, 3=Scissors): 2**

**You throw Paper**

**The computer throws Paper**

**No one wins. Score – you: 1 Score – computer: 0**

**Enter your throw (1=Rock, 2=Paper, 3=Scissors): 1**

**You throw Rock**

**The computer throws Paper**

**You lose Score – you: 1 Score – computer: 1**

**Enter your throw (1=Rock, 2=Paper, 3=Scissors): 1**

**You throw Rock**

**The computer throws Scissors**

**You WIN! Score – you: 2 Score – computer: 1**

**You won the match!**

***Thanks for using Grade Converter***

**Program Checklist**

|  |  |  |
| --- | --- | --- |
|  | **Description** | **Marks** |
| **Knowledge** | **Programming Concepts:**   * Appropriate declaration of variables with meaningful names and  suitable data types * Appropriate use of object classes for input and output * Appropriate use of arithmetic operators to perform program calculations * Appropriate use of methods , objects and classes * Appropriate use of arrays * Appropriate use of conditions & loops |  |
| **Thinking** | **Algorithms:**   * IPO Chart, pseudo code, or flowchart provides detailed step-by-step instructions to properly implement the program specifications. |  |
| **Communication** | **Method documentation:**   * Uses pre and post method documentation   **Program Header:**   * contains programmer’s name, course code, date program written, program name and a comprehensive description of the purpose of the program   **Internal Documentation:**   * comments are used appropriately within the program and provide a meaningful summary of major processes   **Formatting:**   * program source code is properly indented where required and contains appropriate white space for readability * User interface is courteous, esthetically pleasing, and free of spelling and grammar errors |  |
| **Application** | **Implementation:**   * Output is formatted exactly as displayed in the sample layout. * User input is formatted as specified * Program source code is efficient and executes as required with no syntax or logic errors * Error checking logic is used on user input * Error debugging is illustrated * **Bonus**: up to 1 additional bonus mark for improving the user interface beyond what was provided in the specification |  |