**PROGRAMMING PROJECT Name**

Note: Select ONE of the three problems. Clearly indicate your choice. No extra credit will be given for any efforts on the other two problems. All calculations and logic must be done by the program. Do your own programming; do not consult other students, family members, fellow employees, friends. Project is due at the Final Exam. Your program must be demonstrated for the Instructor. Partial credit will be given. Each problem includes bonus points in excess of the 150 point base grade.

(150 pts.)

1. deck of playing cards has 52 cards: A, 2, 3, 4, 5, 6, 7, 8. 9,10, J, Q, K in each suit of clubs, spades, diamonds, and hearts. In the game of 21, the player closest to 21 points without exceeding 21 wins. Cards 2-10 have their face (pip) value. J, Q, and K each count as 10 points. The A can be 1 or 11 points as the player chooses. Each player is dealt two cards face up. Then as many more as desired, one player at a time, starting at the dealer’s left.

You will play the game against the House plus four other players. Label the six players as House, Playerl, Player2, Player3, Player4, Me. Create a graphic for each card dealt, such as a rectangle of the proper color (clubs, spades = black; diamonds, hearts = red) with the suit and value number/letter in the rectangle, or you may prefer to get card images from the Internet.

Each player should have the capacity for six cards dealt. Use a random number generator to determine each card dealt to a player. Note that the same card cannot be dealt twice. Below each player’s cards include two radio buttons labeled “Hit” and “Stand”. The House docs not have the radio buttons but instead will stand if dealt “Blackjack” (Ace plus 10-count card) in the first two cards, or when the House’s point total is a “hard” 17 (Ace counted as 11 unless it puts the house over 21), or above. If the House point total is 16 or less, the House must take another card.

Once a player Stands, your program moves to the next player to the left. If any player’s total exceeds 21, your program indicates that player is “busted” and the player receives no more cards that hand. When all players are done, your program determines and announces the winner, or a tie if one exists, plus the score of the winning total for that hand. Grading for Problem #1:

Activity Maximum Points

Graphics- 45 pts.

Random number card generation/card selection- 35 pts.

Calculations to determine player points, winner, ties, busted- 25 pts.

Sequencing of players- 25 pts.

Answering on-the-spot questions about your program operation- 10 pts.

Complete program listing- 15 pts.

Program comments and other documentation- 15 pts.

Bonus for completing your project before the due date- 10 pts.

TOTAL 180 pts.

Hints:

Devise a working algorithm before doing extensive code development.

Make use of available open class hours.

Start your program design as soon as possible.

Test your program extensively. Address any errors when encountered.

Comment your program to help you answer verbal questions about it.

(150 pts.)

1. YOU are the Supreme Commander of the Students Will Fix the World When We’re Not So Busy army. Your task is to get across the 3100 mile wide Sahara Desert. The only mode of transportation available to you is by Jeep. You have at your disposal as many Jeeps as you want and as much fuel as you need. However, there are no fuel depots within the desert, no time to build any, and no method to store gas on a Jeep other than the gas tank. Each Jeep has a special 45 gallon fuel tank and will get 30.7 mpg. A Jeep cannot run on any other fuel than gasoline nor can the engine be replaced with any alternate means of power. A Jeep cannot tow any other vehicle, nor can you transport a Jeep via truck, ship, plane, etc. You cannot abandon any personnel or Jeep in the desert, nor store any Jeep in the desert. How will you cross the desert ? Write the Visual Basic program to provide the answer.

Your program should use a rectangle or something more creative to show the movement of any Jeep. Divide the screen into 31 vertical sections and have any Jeep travel 100 miles every 2 seconds; show this graphically on the screen, updating every 2 seconds in real time.

Include a solution description in your comments. Grading for Problem #2:

Activity Maximum Points

Graphics- 45 pts.

Real time delay between graphic updates- 35 pts.

Calculations to determine the required intermediate data points- 25 pts. Development and documentation of working algorithm- 25 pts.

Answering on-the-spot questions about your program operation- 10 pts. Complete program listing- 15 pts.

Program comments and other documentation- 15 pts.

Bonus for completing your project before the due date- 10 pts.

TOTAL 180 pts.

Hints:

Devise a working algorithm before doing extensive code development.

Make use of available open class hours.

Start your program design as soon as possible.

Test your program extensively. Address any errors when encountered. Comment your program to help you answer verbal questions about it.

(150 pts.)

1. Refer to the attached drawing. A railroad operates 14 freight train routes each day. Each stop is uniquely numbered as shown in the drawing. Where two routes cross, a junction exists such that one train can go from one route to the other route. This allows a train to go from any starting point to any ending point.

Your task is to create a Visual Basic program that requests the starting point and the ending point, and then determines a detailed path for a train to go from any starting point to any ending point. Your program must not utilize any additional input, and must print the resulting path (including all intermediate points) to the screen in a readable format. For any starting or ending point outside of the valid 0-60 range, re-prompt for a correct entry. For a route with the same starting and ending points (e.g., 39, 39), your program may print the point once or twice (start and end), as you choose. While the shortest route for each train is ideal, it is not a specific requirement of your program. Thus any valid path from starting point to ending point is acceptable. However, a route cannot backtrack on itself, nor omit points within the route, nor include points outside of a valid route, nor print route points out of order for the route.

Your program must also graphically show the attached drawing by using a color-coded circle for each point on the specified 14 routes, labeling the number of each point, and labeling the numbered route. Use a black square for each junction point. Do not use radio buttons or check boxes to substitute for the graphic circles or squares. There is no requirement to “connect the dots” of the routes per the drawing, however each route’s number label and point numbers must be of the same color, and each of the 14 routes must be color-coded with a unique color that is not confused with the color of any of the other 14 routes. There is also no requirement to animate or otherwise graphically show a computed route from start to end point. Grading for Problem #3:

Activity Maximum Points

Graphics- 45 pts.

Program working for at least one non-trivial forward route- 20 pts.

Program working for at least one non-trivial reverse route- 25 pts.

Program working correctly for numerous routes tested- 40 pts.

Answering on-the-spot questions about your program operation- 10 pts. Complete program listing- 15 pts.

Program comments and other documentation- 15 pts.

* Bonus for completing your project before the due date- 10 pts.

TOTAL 180 pts.

Hints:

Devise a working algorithm before doing extensive code development.

Make use of available open class hours.

Start your program design as soon as possible.

Test your program extensively. Address any errors when encountered.

Comment your program to help you answer verbal questions about it.

