1. An overview of the program that answers these questions: What is the robot/program for? What capabilities will it have? How will it be used? (1 paragraph)

This program will be able to have two players play the game of pig. Pig is a game that is played with two dice, players take turns rolling the two dice trying to accumulate the most points and the first player to 100 points win. However, if the player rolls a one the total amount of points that they have accumulated during the round is reduced to zero and it is the next player’s turn. When playing two dice pig if double ones are rolled that player’s total score is reduced to zero.

1. A set of diagrams or drawings of what the robot/program will look like when it is running. They can be hand-drawn, or made in a drawing program, or they can be screenshots of a prototype you made. The diagrams should be labelled and explained. (No length requirement - make it long enough that it is obvious how the program will work from the user’s perspective.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| hello welcome to the game of pig! This game works by having two players take turns rolling two dice, you will be trying to accumulate the most points by either rolling the dice again or holding the dice and the first player to 100 points wins. However, if you roll a one the total amount of points that they have accumulated during the round is reduced to zero and it is the next player’s turn. When playing pig if double ones are rolled that player’s total score is reduced to zero.  Player 1 please enter your name  Joe  Player 2 please enter your name  Bob | Joe’s role  1 and 6  oh no! You rolled a one! Your score for this round is zero! Better luck next time | Bob’s roll  8 and 3  11, would you like to roll again(R) or hold (H)   |  | | --- | | H  Your total score is 11 |   or   |  | | --- | | R  5 and 2  18, would you like to roll again?  H  Your total score is 18 | |

1. A top-down design. You should identify the methods that you will need to write, and what each of these methods will do. List the method headers with their explanations (you should specify a return type and the parameter list for each method).
   1. Main, will take input from the user.
   2. Roll the dice and find the total, public static int roll ()
   3. Check for ones, public static boolean ones(int roll)
   4. Check if they won, public static winners boolean(int totalroll)
   5. Drawing images, public static void draw(int roll)
2. A flowchart or pseudocode for the main method. This should be as complete as possible and should show how the other methods identified in the top-down design will be called.

Int max score = 100

String player1name

String player2name

Int roll

String decision

Pics[]

Print “hello welcome to the game of pig! This game works by having two players take turns rolling two dice, you will be trying to accumulate the most points by either rolling the dice again or holding the dice and the first player to 100 points wins. However, if you roll a one the total amount of points that they have accumulated during the round is reduced to zero and it is the next player’s turn. When playing pig if double ones are rolled that player’s total score is reduced to zero.”

Print “ player1 please enter your name”

player1name= sc.nextline()

Print “ player2 please enter your name”

player2name= sc.nextline()

while( player1total!= maxscore || player2total!= maxscore)

{

Do{

Print “player one’s turn”

Int roll = roll()

If ones() = true{

print “oh no! You rolled a one! Your score for this round is zero!”

Break;}

Else{

Print “player one rolled a” +roll() “ would you like to hold (H) or roll again (R) ”

Decision = sc.nextline()

if( decision. equalsIgnoreCase (“H”){

Totalscore(roll)

break;

{

Else if(decision.equalsIgnoreCase (“R”){

Totalscore(roll)

roll()

}

}while totalscore()<100

Do{

Print “player two’s turn”

Int roll = roll()

If ones() = true{

print “oh no! You rolled a one! Your score for this round is zero!”

Break;}

Else{

Print “player two rolled a” +roll() “ would you like to hold (H) or roll again (R) ”

Decision = sc.nextline()

if( decision. equalsIgnoreCase (“H”){

Totalscore(roll)

break;

{

Else if(decision.equalsIgnoreCase (“R”){

Totalscore(roll)

roll()

}while ones()!= true

}while totalscore()<100