

```

start
  Declarations:
    final num VICTORY_SCORE = 100
    final num twoOnesRolled = -2000
    final num oneOneRolled = -200
    final num doubleScore = 10
    final num doubleSixScore = 12

  main
    Declarations:
      final num playerOneScore = 0
      final num playerTwoScore = 1
      final num CPUScore = 2
      boolean opponentSelection = opponentSelection()
      String[]playerNames = getNames(opponentSelection)
      num[]scores = new num[3]
      boolean gameIsOver = false
      num roundNum = 1
      printInstructions()
      while gameIsOver = false
        printRoundInfo(roundNum, true, playerNames,
scores, true)
        scores[playerOneScore] += logicLoop(true,
false, scores[playerOneScore], playerNames)
        if scores[playerOneScore] < -1800
          scores[playerOneScore] = 0
        endif
        if scores[playerOneScore] >= VICTORY_SCORE
          gameIsOver = true
          output playerNames[0] + " won!"
        endif
        wait(1000)
        if opponentSelection = true
          scores[CPUScore] += logicLoop(true,
true, scores[CPUScore], playerNames)
          if scores[CPUScore] < -1800
            scores[CPUScore] = 0
          endif
          if scores[CPUScore] >= VICTORY_SCORE
            gameIsOver = true
            output "The computer won!"
          endif
        else
          printRoundInfo(roundNum,
false, playerNames, scores, true)
          roundNum++
        endif
      else
        scores[playerTwoScore] +=
logicLoop(false, true, scores[playerTwoScore], playerNames)
        if scores[playerTwoScore] < -1800
          scores[playerTwoScore] = 0
        endif
      endif
    endwhile
  endwhile
endwhile

```

```

                                if scores[playerTwoScore] >=
VICTORY_SCORE
                                gameIsOver = true
                                output playerNames[1] + "
won!"
                                else
                                printRoundInfo(roundNum,
false, playerNames, scores, false)
                                roundNum++
                                endif
                                endif
                                endif
stop

```

QUESTION do declarations have to be done if they're being defined in parameters?

```

void printRoundInfo(num roundNum, boolean startOfRound,
String[]playerNames, int[]scores, boolean opponentChoice)
    if startOfRound
        output "-----Round " + roundNum + "-----"
        return
    endif
    if opponentChoice
        wait(1000)
        output "Round " + roundNum + " scores:"
        wait(1000)
        output playerNames[0] + ": " + scores[0]
        wait(1000)
        output "CPU: " + scores[2]
        wait(1000)
    endif
    else
        wait(1000)
        output "Round " + roundNum + " scores:"
        wait(1000)
        output playerNames[0] + ": " + scores[0]
        wait(1000)
        output playerNames[1] + ": " + scores[1]
        wait(1000)
    endif
return
void printInstructions()
    output "-----PIG-----"
    wait(1000)
    output "Two players (or one player and a computer) will take turns
rolling die."
    wait(1000)
    output "If no 1 appears, the sum of the die will be added to a
running total for the turn."
    wait(1000)

```

```

        output "The player can then choose to pass the turn, adding their
running total to their bank, or roll again."
        wait(1000)
        output "For each turn after the first, 2 times the roll number
minus 1 ( $2 * (n-1)$ ) will be added as a bonus."
        wait(1000)
        output "If doubles are rolled, 10 points will be added, or 12 if
two sixes are rolled."
        wait(1000)
        output "If a 1 appears, the running total is reset to 0. If two
1's appear, the bank total is reset. The round ends after both of these."
        wait(1000)
        output "The first player to reach 100 banked points wins."
        wait(1000)
    return

```

```

String[] getNames(boolean onePlayer)
    Declarations:
        String[] playerNames = new String[2]
        String confirmation
        output "Enter username for Player 1: "
        input playerNames[0]
        output "You entered: " + playerNames[0] + ". Type 'no' if
this is incorrect or anything else to continue: "
        input confirmation
        while confirmation = "no"
            output "Enter username for Player 1: "
            input playerNames[0]
            output "You entered: " + playerNames[0] + ".
Type 'no' if this is incorrect or anything else to continue: "
            input confirmation
        endwhile
        if onePlayer = false
            output "Enter username for Player 2: "
            input playerNames[1]
            output "You entered: " + playerNames[1] + ".
Type 'no' if this is incorrect or anything else to continue: "
            input confirmation
            while confirmation = "no"
                output "Enter username for Player 2: "
                input playerNames[1]
                output "You entered: " + playerNames[1]
+ ". Type 'no' if this is incorrect or anything else to continue: "
                input confirmation
            endwhile
        endif
    return playerNames

```

```

boolean opponentSelection()
    Declarations:
        String playerChoice
        output "This game allows you to either pass the device
between two users, or to play against a computer opponent."

```

```

        output "If you would like to play against another player, type
\"Player\". \nIf you would like to play against a computer, type
'Computer'"
        input playerChoice
        while playerChoice != "Player" && playerChoice !=
"Computer"
            output "Invalid input."
            output "If you would like to play against another player, type
\"Player\". \nIf you would like to play against a computer, type
'Computer': "
            input playerChoice
        endwhile
        return !playerChoice.equalsIgnoreCase("Player")

num rollDice()
    Declarations:
        num[] rollValues = new num[2]
        num rollValues[2] = random(0,7), random(0,7)
    return rollValues[]

num scoringSystem(num score, num[] rollValues, num bonusMultiplier,
num bankedScore)
    Declarations:
        num subtotal

        if rollValues[0] = 1 && rollValues[1] = 1
            output "Two 1's were rolled. The round ends
and the banked score is reset to 0."
            score = twoOnesRolled
            wait(1000)
            return score
        endif
        if rollValues[0] = 1 || rollValues[1] = 1
            output "A " + rollValues[0] + " and a " +
rollValues[1] + " were rolled. The round ends, no score banked.
            score = oneOneRolled
            wait(1000)
            return score
        endif
        if rollValues[0] = rollValues[1] && rollValues[0] = 6
            subtotal = doubleSixScore + (2 *
bonusMultiplier)
            score += subtotal
            output "Two 6's were rolled. " + subtotal + "
added to total, which is now
            " + score + ". Banked total remains " +
bankedScore
            return score
        endif
        if rollValues[0] = rollValues[1] && rollValues[0] &&
rollValues[0] != 6
            subtotal = doubleScore + (2 *
bonusMultiplier)
            score += subtotal

```

```

        output "Two " + rollValues[1] + "'s were
rolled. " + subtotal + " added to total, which is now "
        + score + ". Banked total remains " +
bankedScore
        wait(1000)
        return score
    endif
    subtotal = rollValues[0] + rollValues[1] + (2 *
bonusMultiplier)
    score += subtotal
    output "A " + rollValues[0] + " and a " + rollValues[1] +
" were rolled. " + subtotal + " added to total, which is now "
    + score + ". Banked total remains " + bankedScore
    wait(1000)
    return score

    boolean playerChoice()
        Declarations:
            String playerChoice
            output "Would you like to bank your score or
roll again with an additional bonus multiplier? \n
            "Type 'bank' to bank or 'roll' to roll again:
            "
            input playerChoice
            while playerChoice != bank && playerChoice !=
roll
                output "Invalid input."
                output "Would you like to bank your
score or roll again with an additional bonus multiplier? \n
                "Type 'bank' to bank or 'roll' to roll
again: "
                input playerChoice
            endwhile
            return playerChoice = roll

    num logicLoop(boolean opponentSelection, boolean isSecondTurn, num
bankScore, String[]playerNames)
        Declarations:
            num score
            num bonusMultiplier
            num coinFlip
            if isSecondTurn = false
                output playerNames[0] + "'s turn"
                score = scoringSystem(score, rollDice(),
bonusMultiplier, bankScore)
                if score <= 0 && score > -1901
                    return 0
                endif
                if score <= -1901
                    return twoOnesRolled
                endif
                while playerChoice() = true
                    bonusMultiplier++

```

```

        score = scoringSystem(score,
rollDice(), bonusMultiplier, bankScore)
        if score <= 0 && score > -1901
            return 0
        endif
        if score <= -1901
            return twoOnesRolled
        endif
        if score >= VICTORY_SCORE
            return score
        endif
    endwhile
    output playerNames[0] + " banked " + score +
" points, bringing their total to " + (bankscore + score)
    return score
endif
if opponentSelection = true
    output "Computer's turn: "
    score = scoringSystem(score, rollDice(),
bonusMultiplier, bankScore)
    if score <= 0 && score > -1901
        return 0
    endif
    if score <= -1901
        return twoOnesRolled
    endif
    coinFlip = random(0,2)
    while coinFlip = 0
        output "The computer chose to roll
again."
        score = scoringSystem(score,
rollDice(), bonusMultiplier, bankScore)
        if score <= 0 && score > -1901
            return 0
        endif
        if score <= -1901
            return twoOnesRolled
        endif
        if score >= VICTORY_SCORE
            return score
        endif
        wait(1000)
        bonusMultiplier++
        coinFlip = random(0,2)
    endwhile
    output "The computer chose to bank " + score
+ " points, bringing their total to " + (bankscore + score)
    return score
endif
else
    output playerNames[1] + "'s turn"
    score = scoringSystem(score, rollDice(),
bonusMultiplier, bankScore)
    if score <= 0 && score > -1901
        return 0
    endif

```

```

endif
if score <= -1901
    return twoOnesRolled
endif
while playerChoice() = true
    bonusMultiplier++
    score = scoringSystem(score,
rollDice(), bonusMultiplier, bankScore)
    if score <= 0 && score > -1901
        return 0
    endif
    if score <= -1901
        return twoOnesRolled
    endif
    if score >= VICTORY_SCORE
        return score
    endif
endwhile
output playerNames[1] + " banked " + score +
" points, bringing their total to " + (bankscore + score)
return score
endif
return score

```