

**CM-1601 Programming Fundamentals Coursework Part B .**

Imandi Karunarathna – (20200306)

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Scenario | Input From User | Robot 1 Output | Robot 2 Output | Robot 3 Output | Expected Result | Code |
| Same Suit | K of Clubs | 7 of Clubs | A of Clubs | 10 of Clubs | Robot 2 Wins | (1) |
| Different Suit | 10 of Spades | J of Hearts | Q of Clubs | 7 of Diamonds | User wins | (2) |
| Robots put Trumps (Spades) | 10 of Clubs | 7 of Spades | 10 of Spades | J of Spades | Robot 3 wins | (3) |
| 2 Robots put Trumps (Spades) | 7 of Hearts | 10 of Spades | 8 of Spades | 8 of Diamonds | Robot 1 wins | (4) |
| 1 Robot put Trumps | 7 of Hearts | 7 of Spades | 8 of Clubs | 10 of Hearts | Robot 2 wins(T=Clubs) | (5) |

Code of the Test Case 1-

if ((suitRobo3.equals(suitUser)) && (suitRobo3.equals(suitRobo1)) && (suitRobo3.equals(suitRobo2))) {  
 if ((robo3Digit > userDigit) && (robo3Digit > robo1Digit) && (robo3Digit > robo2Digit)) {  
 startGame = "robo3"**;** System.*out*.println("the third computer wins")**;** robo3Points += 1**;** } else if ((robo1Digit > robo3Digit) && (robo1Digit > robo2Digit) && (robo1Digit > userDigit)) {  
 startGame = "robo1"**;** System.*out*.println("the first computer wins")**;** robo1Points += 1**;** } else if ((robo2Digit > robo3Digit) && (robo2Digit > robo1Digit) && (robo2Digit > userDigit)) {  
 startGame = "robo2"**;** System.*out*.println("the second computer wins")**;** robo2Points += 1**;** } else {  
 startGame = "user"**;** System.*out*.println("You win ")**;** userPoints += 1**;** }

Test Case 2 –

else {

startGame = "user"**;**System.*out*.println("You win")**;**userPoints += 1**;**

Test Case 3-

else if ((suitUser.equals(cardTrick.trumpSuit)) && (suitRobo1.equals(cardTrick.trumpSuit)) && (suitRobo2.equals(cardTrick.trumpSuit))) {  
 if ((userDigit > robo1Digit) && (userDigit > robo2Digit)) {  
 startGame = "user"**;** System.*out*.println("You win ")**;** userPoints += 1**;** } else if ((robo1Digit > userDigit) && (robo1Digit > robo2Digit)) {  
 startGame = "robo1"**;** System.*out*.println("the first computer wins ")**;** robo1Points += 1**;** } else {  
 startGame = "robo2"**;** System.*out*.println("the second computer wins ")**;** robo2Points += 1**;** }

Test Case 4-

else if ((suitUser.equals(cardTrick.trumpSuit)) && (suitRobo1.equals(cardTrick.trumpSuit))) {  
 if ((userDigit > robo1Digit)) {  
 startGame = "user"**;** System.*out*.println("You win ")**;** userPoints += 1**;** } else {  
 startGame = "robo1"**;** System.*out*.println("the first computer wins")**;** robo1Points += 1**;** }

Test Case 5-

else if (suitRobo1.equals(cardTrick.trumpSuit)) {  
 startGame = "robo1"**;** System.*out*.println("the first computer wins")**;** robo1Points += 1**;**

Main Class

import java.util.ArrayList**;**import java.util.Scanner**;**import java.util.Random**;**public class MainClass {  
  
  
 public static void main(String[] args) {  
 Random r= new Random()**;** Scanner input = new Scanner(System.*in*)**;** String leadTrump = "User"**;** //initialises trump lead as user  
 String startGame = "user"**;** //game is started by the user  
 CardDeck cardDeck = new CardDeck()**;** //instance of CardDeck created  
 Skill cardTrick = new Skill()**;** //instance of Skill created  
 String userStart = null**;** //userStart initialised to null  
  
  
  
  
 Cards card1 = new Cards("♦"**,** "7"**,** 7)**;** //card 1 stores 7 0f diamonds  
 Cards card2 = new Cards("♦"**,** "8"**,** 8)**;** //card 2 stores 8 0f diamonds  
 Cards card3 = new Cards("♦"**,** "9"**,** 9)**;** //card 3 stores 9 0f diamonds  
 Cards card4 = new Cards("♦"**,** "10"**,** 10)**;** //card 4 stores 10 0f diamonds  
 Cards card5 = new Cards("♦"**,** "J"**,** 11)**;** //card 5 stores jack 0f diamonds  
 Cards card6 = new Cards("♦"**,** "Q"**,** 12)**;** //card 6 stores queen 0f diamonds  
 Cards card7= new Cards("♦"**,** "K"**,** 13)**;** //card 7 stores king 0f diamonds  
 Cards card8 = new Cards("♦"**,** "A"**,** 14)**;** //card 8 stores ace of diamonds  
  
 Cards card9 = new Cards("♥"**,** "7"**,** 7)**;** //card9 stores the 7 of hearts  
 Cards card10 = new Cards("♥"**,** "8"**,** 8)**;** //card10 stores the 8 of hearts  
 Cards card11 = new Cards("♥"**,** "9"**,** 9)**;** //card11 stores the 9 of hearts  
 Cards card12 = new Cards("♥"**,** "10"**,** 10)**;** //card12 stores the 10 of hearts  
 Cards card13 = new Cards("♥"**,** "J"**,** 11)**;** //card13 stores the jack of hearts  
 Cards card14 = new Cards("♥"**,** "Q"**,** 12)**;** //card14 stores the queen of hearts  
 Cards card15 = new Cards("♥"**,** "K"**,** 13)**;** //card15 stores the king of hearts  
 Cards card16 = new Cards("♥"**,** "A"**,** 14)**;** //card16 stores the ace of hearts  
  
  
 Cards card17 = new Cards ("♠"**,** "7"**,** 7)**;** //card17 stores the 7 of spades  
 Cards card18 = new Cards("♠"**,** "8"**,** 8)**;** //card18 stores the 8 of spades  
 Cards card19 = new Cards("♠"**,** "9"**,** 9)**;** //card 19 stores the 9 of spades  
 Cards card20 = new Cards("♠"**,** "10"**,** 10)**;** //card20 stores the 10 of spades  
 Cards card21 = new Cards("♠"**,** "J"**,** 11)**;** //card21 stores the jack of spades  
 Cards card22 = new Cards("♠"**,** "Q"**,** 12)**;** //card22 stores the queen of spades  
 Cards card23 = new Cards("♠"**,** "K"**,** 13)**;** //card23 stores the king of spades  
 Cards card24 = new Cards("♠"**,** "A"**,** 14)**;** //card24 stores the ace of spades  
  
  
  
 Cards card25 = new Cards("♣"**,** "7"**,** 7)**;** //card 25 stores the 7 of clubs  
 Cards card26 = new Cards("♣"**,** "8"**,** 8)**;** //card 26 stores the 8 of clubs  
 Cards card27 = new Cards("♣"**,** "9"**,** 9)**;** //card 27 stores the 9 of clubs  
 Cards card28 = new Cards("♣"**,** "10"**,** 10)**;** //card 28 stores the 10 of clubs  
 Cards card29 = new Cards("♣"**,** "J"**,** 11)**;** //card29 stores the jack of clubs  
 Cards card30 = new Cards("♣"**,** "Q"**,** 12)**;** //card30 stores the queen of clubs  
 Cards card31 = new Cards("♣"**,** "K"**,** 13)**;** //card31 stores the king of clubs  
 Cards card32 = new Cards("♣"**,** "A"**,** 14)**;** //cad32 stores the ace of clubs  
  
  
  
  
  
  
 cardTrick.addCards(card2**,**card3**,**card9**,**card6**,**card12**,**card17**,**card21**,**card30**,**card24**,**card5**,**card27**,**card14**,**card19**,**card32**,**card8**,**card23**,**card1**,**card11**,**card29**,**card20**,**card16**,**card21**,**card4**,**card26**,**card13**,**card15**,**card28**,**card18**,**card10**,**card25**,**card7)**;** //this line has a jumbled version of the 32 cards  
  
 System.*out*.println("Welcome to OMI!")**;** int playagain = 1**;** //player again initialised to 1  
 while (playagain != 2) { //while playeragain value not equal 2  
  
  
 int userPoints = 0**,** robo1Points = 0**,** robo2Points = 0**,** robo3Points = 0**;** //initialises the users,first computer,second computer,third computer points to zero  
 cardTrick.shuffleDeck(cardTrick.getCardDeck())**;** //shuffles the 32 cards  
  
  
 if (leadTrump.equals("User")) { //checks if trump is lead by user  
  
 startGame = "user"**;** //game is started by user  
 cardTrick.trumpUser(cardTrick.getUser())**;** //gets the trump suit from user  
  
 } else if (leadTrump.equals("Robo1")) { //checks if trump is lead by first computer  
 cardTrick.leadRobo1()**;** //starts dividing cards equally starting from first computer  
 for (int x = 0**;** x <= cardTrick.getUser().size() - 1**;** x++) {  
 System.*out*.println("Your cards are : " + cardTrick.getUser().get(x).getNumber() + cardTrick.getUser().get(x).getSuit())**;** //prints users cards  
 }  
 cardTrick.roboChoiceTrump(cardTrick.getRobo1())**;** //gets trump from robo  
  
 } else if (leadTrump.equals("Robo2")) { //checks if trumps lead by robo2  
 cardTrick.leadRobo2()**;** //starts dividing cards equally starting from robo2  
 for (int x = 0**;** x <= cardTrick.getUser().size() - 1**;** x++) {  
 System.*out*.println("Your cards are : " + cardTrick.getUser().get(x).getNumber() + cardTrick.getUser().get(x).getSuit())**;** //prints users card  
 }  
 cardTrick.roboChoiceTrump(cardTrick.getRobo2())**;** //gets trump from robo  
 } else {  
 cardTrick.leadRobo3()**;** //else trump is lead by robo 3 and cards are going to get divided equally starting from robo3  
 for (int x = 0**;** x <= cardTrick.getUser().size() - 1**;** x++) {  
 System.*out*.println("Your cards are : " + cardTrick.getUser().get(x).getNumber() + cardTrick.getUser().get(x).getSuit())**;** //prints users cards  
 }  
 cardTrick.roboChoiceTrump(cardTrick.getRobo3())**;** //gets trump from robo  
 }  
 for (int gameRound = 1**;** gameRound < 8**;** gameRound++) { //for loop which goes for the 8 rounds in the game  
 System.*out*.println("This is round" + " " + gameRound)**;** //prints what is the current round  
 int randomRobo1 = 0**,** randomRobo2 = 0**,** randomRobo3 = 0**,** randomUser = 0**;** //initialises the random robo1,robo2,robo3 and user to 0  
 int userDigit = 0**;** //initialises the users digit to 0  
 int robo1Digit = 0**;** //initialises the robo1 digit to 0  
 int robo2Digit = 0**;** //initialises the robo2 digit to 0  
 int robo3Digit = 0**;** //initialises the robo 3 digit to 0  
 String suitRobo1 = null**,** suitRobo2 = null**,** suitRobo3 = null**,** suitUser = null**;** //initialises the robo1,robo2,robo3 and users suit to null  
 if (startGame.equals("user")) { //below if will run if user has started the game  
  
  
 System.*out*.println("helloo please enter your card: ")**;** userStart = input.nextLine()**;** //users card is obtained from user  
 int z = 0**;** for (int i = 0**;** i < cardTrick.user.size()**;** i++) {  
 if ((cardTrick.getUser().get(i).getNumber() + cardTrick.getUser().get(i).getSuit()).equals(userStart)) {  
 z += i**;** } //prints users cards  
 }  
  
 userDigit = cardTrick.user.get(z).getDigit()**;** //assigns users current index digit to userDigit  
 suitUser = cardTrick.user.get(z).getSuit()**;** //assigns users current index suit to userSuit  
  
 ArrayList<Cards> robo1Arr = new ArrayList<>()**;** //temporary array list of robo1  
 for (int y = 0**;** y < cardTrick.user.size() **;** y++) { //goes in a for loop of the user cards  
 if (cardTrick.user.get(z).getSuit().equals(cardTrick.robo1.get(y).getSuit())) { //checks if users suit equals robo suit  
 robo1Arr.add(cardTrick.robo1.get(y))**;** //if found a sim ilar suit card will get added to temporarry array  
 }  
 }  
  
 ArrayList<Cards> robo2Arr = new ArrayList<>()**;** //temporary array list of robo2  
 for (int w = 0**;** w < cardTrick.robo2.size()**;** w++) { //goes in a for loop of the robo2 cards  
 if (cardTrick.user.get(z).getSuit().equals(cardTrick.robo2.get(w).getSuit())) { //checks if users suit equals robo2 suit  
 robo2Arr.add(cardTrick.robo2.get(w))**;** //if found a sim ilar suit card will get added to temporarry array  
 }  
 }  
  
  
 ArrayList<Cards> robo3Arr = new ArrayList<>()**;** //temporary array list of robo3  
 for (int w = 0**;** w < cardTrick.robo3.size()**;** w++) { //goes in a for loop of the robo3 cards  
 if (cardTrick.user.get(z).getSuit().equals(cardTrick.robo3.get(w).getSuit())) { //checks if users suit equals robo3 suit  
 robo3Arr.add(cardTrick.robo3.get(w))**;** //if found a sim ilar suit card will get added to temporarry array  
  
 }  
 }  
 if (robo1Arr.size() > 0) { //if found a card in robo 1 with similar suit as user  
 randomRobo1 = r.nextInt(robo1Arr.size())**;** //a random index is calculated  
 Cards cardRobo1 = robo1Arr.get(randomRobo1)**;** //adds that calculate index card to temporaray cardrobo1  
 robo1Digit = cardRobo1.getDigit()**;** //gets random card digit  
 suitRobo1 = cardRobo1.getSuit()**;** //gets random card suit  
 cardTrick.robo1.remove(cardRobo1)**;** //removes random card  
 System.*out*.println("first computer puts card : " + " " + cardRobo1.getNumber() + cardRobo1.getSuit())**;** //displays card to be removed on to console  
  
  
 } else {  
 randomRobo1 = r.nextInt(cardTrick.robo1.size())**;** //calculates a random card  
 Cards cardRobo1 = cardTrick.robo1.get(randomRobo1)**;** //adds that card to random cardrobo1  
 robo1Digit = cardRobo1.getDigit()**;** //gets random card digit  
 suitRobo1 = cardRobo1.getSuit()**;** //gets random card suit  
 cardTrick.robo1.remove(cardRobo1)**;** //removes random card  
 System.*out*.println("first computer puts card : " + " " + cardRobo1.getNumber() + cardRobo1.getSuit())**;** //displays card to be removed on to console  
  
 }  
  
 if (robo2Arr.size() > 0) { //if found a card in robo 2 with similar suit as user  
 randomRobo2 = r.nextInt(robo2Arr.size())**;** //a random index is calculated  
 Cards cardRobo2 = robo2Arr.get(randomRobo2)**;** //adds that calculate index card to temporaray cardrobo2  
 robo2Digit = cardRobo2.getDigit()**;** //gets random card digit  
 suitRobo2 = cardRobo2.getSuit()**;** //gets random card suit  
 cardTrick.robo2.remove(cardRobo2)**;** //removes random card  
 System.*out*.println("second computer puts card : " + " " + cardRobo2.getNumber() + cardRobo2.getSuit())**;** //displays card to be removed on to console  
  
 } else {  
 randomRobo2 = r.nextInt(cardTrick.robo2.size())**;** //calculates a random card  
 Cards cardRobo2 = cardTrick.robo2.get(randomRobo2)**;** //adds that card to random cardrobo2  
 robo2Digit = cardRobo2.getDigit()**;** //gets random card digit  
 suitRobo2 = cardRobo2.getSuit()**;** //gets random card suit  
 cardTrick.robo2.remove(cardRobo2)**;** //removes random card  
 System.*out*.println("second computer puts card : " + " " + cardRobo2.getNumber() + cardRobo2.getSuit())**;** //displays card to be removed on to console  
  
 }  
  
  
 if (robo3Arr.size() > 0) { //if found a card in robo 3 with similar suit as user  
 randomRobo3 = r.nextInt(robo3Arr.size())**;** //a random index is calculated  
 Cards cardRobo3 = robo3Arr.get(randomRobo3)**;** //adds that calculate index card to temporaray cardrobo3  
 robo3Digit = cardRobo3.getDigit()**;** //gets random card digit  
 suitRobo3 = cardRobo3.getSuit()**;** //gets random card suit  
 cardTrick.robo3.remove(cardRobo3)**;** //removes random card  
 System.*out*.println("third computer puts card : " + " " + cardRobo3.getNumber() + cardRobo3.getSuit())**;** //displays card to be removed on to console  
  
 } else {  
 randomRobo3 = r.nextInt(cardTrick.robo3.size())**;** //calculates a random card  
 Cards cardRobo3 = cardTrick.robo3.get(randomRobo3)**;** //adds that card to random cardrobo3  
 robo3Digit = cardRobo3.getDigit()**;** //gets random card digit  
 suitRobo3 = cardRobo3.getSuit()**;** //gets random card suit  
 cardTrick.robo3.remove(cardRobo3)**;** //removes random card  
 System.*out*.println("third computer puts card : " + " " + cardRobo3.getNumber() + cardRobo3.getSuit())**;** //displays card to be removed on to console  
  
 }  
  
 if ((suitRobo1.equals(suitRobo2)) && (suitRobo1.equals(suitRobo3)) && (suitRobo1.equals(suitUser))) { //checks if robo1 suit equals robo2,robo3 and users  
 if ((robo1Digit > robo2Digit) && (robo1Digit > robo3Digit) && (robo1Digit > userDigit)) { //checks if robo1 digit greater than robo2,robo3and user  
 startGame = "robo1"**;** //robo1 starts game  
 System.*out*.println("The first computer wins ")**;** //robo1 wins  
 robo1Points += 1**;** //robo1s points incremented  
 } else if ((robo2Digit > robo3Digit) && (robo2Digit > userDigit) && (robo2Digit > robo1Digit)) { //checks if robo2 digit greater than robo3,user,robo1  
 startGame = "robo2"**;** //robo2 starts game  
 System.*out*.println("The second computer wins ")**;** //robo2 wins  
 robo2Points += 1**;** //robo2 points incremented  
 } else if ((robo3Digit > userDigit) && (robo3Digit > robo2Digit) && (robo3Digit > robo1Digit)) { //checks if robo3 digit greater than robo2,user,robo1  
 startGame = "robo3"**;** //robo3 starts game  
 System.*out*.println("The third computer wins ")**;** //robo3 wins  
 robo3Points += 1**;** //robo3 points incremented  
 } else {  
 startGame = "user"**;** //user starts game  
 System.*out*.println("You win ")**;** //user wins  
 userPoints += 1**;** //users points incremented  
 }  
 } else if ((suitUser.equals(suitRobo1)) && (suitUser.equals(suitRobo2)) && (!suitRobo3.equals(cardTrick.trumpSuit))) { //checks if users suit is same as robo1 and robo2 but robo3 suit not trump  
 if ((userDigit > robo1Digit) && (userDigit > robo2Digit)) { //checks if users digit higher than robo1 and robo 2 digit  
 startGame = "user"**;** //user starts game  
 System.*out*.println("You win")**;** //user wins  
 userPoints += 1**;** //users points are incremented  
 } else if ((robo1Digit > userDigit) && (robo1Digit > robo2Digit)) { //checks if robo1 digit greater than users and robo2  
 startGame = "robo1"**;** //robo1 starts game  
 System.*out*.println("The first computer wins")**;** //robo1 wins  
 robo1Points += 1**;** //robo1 points incremented  
 } else {  
 startGame = "robo2"**;** //robo2 starts game  
 System.*out*.println("The second computer wins")**;** //robo2 wins  
 robo2Points += 1**;** //robo2 points are incremented  
 }  
 } else if ((suitUser.equals(suitRobo2)) && (suitUser.equals(suitRobo3)) && (!suitRobo1.equals(cardTrick.trumpSuit))) { //checks if users card suit is same as robo2 and robo3 but robo1 suit not equal to trump  
 if ((userDigit > robo2Digit) && (userDigit > robo3Digit)) { //checks if users card digit greater robo2 digit and robo3 digit  
 startGame = "user"**;** //user starts game  
 System.*out*.println("You win")**;** //user wins  
 userPoints += 1**;** //usrs points are incremented  
 } else if ((robo2Digit > userDigit) && (robo2Digit > robo3Digit)) { //checks if roobo2 digit greater than users and robo3  
 startGame = "robo2"**;** //robo2 starts game  
 System.*out*.println("the second computer wins")**;** //robo2 wins  
 robo2Points += 1**;** //robo2 points are incrememnted  
 } else {  
 startGame = "robo3"**;** //robo3 starts game  
 System.*out*.println("the third computer wins")**;** //robo3 wins  
 robo3Points += 1**;** //robo3 points are incremnented  
 }  
 } else if ((suitUser.equals(suitRobo1)) && (suitUser.equals(suitRobo3)) && (!suitRobo2.equals(cardTrick.trumpSuit))) { //checks if users suit is same as robo1 and robo3 suit  
 if ((userDigit > robo1Digit) && (userDigit > robo3Digit)) { //checks if users digit higher than robo1 and robo3 digit  
 startGame = "user"**;** //user wins  
 System.*out*.println("you win")**;** userPoints += 1**;** //users points are incremented  
 } else if ((robo1Digit > userDigit) && (robo1Digit > robo3Digit)) { //checks if robo1 digit greater than users digit and robo3 digit  
 startGame = "robo1"**;** //robo1 starts game  
 System.*out*.println("the first computer wins")**;** //robo1 wins  
 robo1Points += 1**;** //robo1 points incremented  
 } else {  
 startGame = "robo3"**;** //robo3 starts game  
 System.*out*.println("the third computer wins")**;** //robo3 wins  
 robo3Points += 1**;** //robo3 points incremented  
 }  
 } else if (suitRobo1.equals(cardTrick.trumpSuit) && (suitRobo2.equals(cardTrick.trumpSuit)) && (suitRobo3.equals(cardTrick.trumpSuit))) { //checks if robo1,robo2,robo3 suits equal trumps  
 if ((robo1Digit > robo2Digit) && (robo1Digit > robo3Digit)) { //checlks if robo1 digit greater than robo2 and robo3 digit  
 startGame = "robo1"**;** //robo1 starts game  
 System.*out*.println("the first computer wins")**;** //robo1 wins  
 robo1Points += 1**;** //robo1 points are incremented  
 } else if ((robo2Digit > robo1Digit) && (robo2Digit > robo3Digit)) { //checks if robo2 digit is higher than robo1 and robo3 digit  
 startGame = "robo2"**;** //robo2 starts game  
 System.*out*.println("the second computer wins")**;** //robo2 wins  
 robo2Points += 1**;** //robo2 points are incremented  
 } else {  
 startGame = "robo3"**;** //robo3 starts game  
 System.*out*.println("the third computer wins")**;** //robo3 wins  
 robo3Points += 1**;** //robo3 poinyts are incremeneted  
 }  
 } else if ((suitRobo1.equals(cardTrick.trumpSuit)) && (suitRobo2.equals(cardTrick.trumpSuit))) { //checks if roob1 and robo2 suits are trumps  
 if (robo1Digit > robo2Digit) { //checks if robo1 digit higher than robo2  
 startGame = "robo1"**;** //robo1 starts game  
 System.*out*.println("the first computer wins")**;** //robo1 wins  
 robo1Points += 1**;** //robo1 points are incremenetd  
 } else {  
 startGame = "robo2"**;** //robo2 starts game  
 System.*out*.println("the second computer wins ")**;** //robo2 wins  
 robo2Points += 1**;** //robo2 points are incremented  
 }  
 } else if ((suitRobo1.equals(cardTrick.trumpSuit)) && (suitRobo3.equals(cardTrick.trumpSuit))) { //checks if robo1 and robo3 suit is trumps  
 if (robo1Digit > robo3Digit) { //checks if robo1 digit higher than robo3  
 startGame = "robo1"**;** //robo1 starts game  
 System.*out*.println("the first computer wins")**;** //robo1 wins  
 robo1Points += 1**;** //robo1 points are incremented  
 } else {  
 startGame = "robo3"**;** //robo3 satrts game  
 System.*out*.println("the third computer wins")**;** //robo3 wins  
 robo3Points += 1**;** //robo3 points are incremented  
 }  
 } else if ((suitRobo2.equals(cardTrick.trumpSuit)) && (suitRobo3.equals(cardTrick.trumpSuit))) { //checks if robo2 and robo3 suit is trumps  
 if (robo2Digit > robo3Digit) { //checks if robo2 digit greater than robo3 digit  
 startGame = "robo2"**;** //robo2 starts game  
 System.*out*.println("the second computer wins")**;** //robo2 wins  
 robo2Points += 1**;** //robo2 points are incremented  
 } else {  
 startGame = "robo3"**;** //robo3 starts game  
 System.*out*.println("the third computer wins")**;** //robo3 wins  
 robo3Points += 1**;** //robo3 points are incremented  
 }  
 } else if (suitRobo1.equals(cardTrick.trumpSuit)) { //checks if robo1 suit is trumps  
 startGame = "robo1"**;** //robo1 starts game  
 System.*out*.println("the first computer wins")**;** //robo1 wins  
 robo1Points += 1**;** //robo1 points are incremented  
 } else if (suitRobo2.equals(cardTrick.trumpSuit)) { //checks if robo2 suit is trumps  
 startGame = "robo2"**;** //robo 2 starts game  
 System.*out*.println("the second computer wins")**;** //robo2 wins  
 robo2Points += 1**;** //robo2 points are incremented  
 } else if (suitRobo3.equals(cardTrick.trumpSuit)) { //checks if robo3 suit is trumps  
 System.*out*.println("the third computer wins")**;** //robo3 wins  
 robo3Points += 1**;** //robo3 points are incremented  
 } else {  
 startGame = "user"**;** //user starts game  
 System.*out*.println("you win")**;** //user wins  
 userPoints += 1**;** //users points have been incremeted  
 }  
  
 cardTrick.user.remove(z)**;** //index card removed from user array list  
 for (int x = 0**;** x <= cardTrick.getUser().size() - 1**;** x++) {  
 System.*out*.println("your cards are : " + cardTrick.getUser().get(x).getNumber() + cardTrick.getUser().get(x).getSuit())**;** //displays remaining cards on console  
 }  
  
  
 } else if (startGame.equals("robo1")) { //checks if robo1 started game  
 randomRobo1 = r.nextInt(cardTrick.robo1.size())**;** //random index assigned to random robo1  
 Cards cardRobo1 = cardTrick.robo1.get(randomRobo1)**;** //random card calculated  
 robo1Digit = cardRobo1.getDigit()**;** //gets random card digit  
 suitRobo1 = cardRobo1.getSuit()**;** //gets random card suit  
 cardTrick.robo1.remove(cardRobo1)**;** //removes random card  
 System.*out*.println("computer one puts in card :" + " " + cardRobo1.getNumber() + cardRobo1.getSuit())**;** //prints computer 1 cards  
  
 ArrayList<Cards> robo2Arr = new ArrayList<>()**;** //temporaray array list robo2arr created  
 for (int w = 0**;** w < cardTrick.robo2.size() - 1**;** w++) { //checks in robo2 if it has a similar card suit to robo1  
 if (cardTrick.robo1.get(randomRobo1).getSuit().equals(cardTrick.robo2.get(w).getSuit())) {  
 robo2Arr.add(cardTrick.robo2.get(w))**;** //adds that card to temporaray list  
  
 }  
 }  
  
 if (robo2Arr.size() > 0) { //if there is a similar card  
 randomRobo2 = r.nextInt(robo2Arr.size())**;** //gets random index  
 Cards cardRobo2 = robo2Arr.get(randomRobo2)**;** //finds random card  
 robo2Digit = cardRobo2.getDigit()**;** //gets random digit  
 suitRobo2 = cardRobo2.getSuit()**;** //gets random suit  
 cardTrick.robo2.remove(cardRobo2)**;** //removes card  
 System.*out*.println("computer 2 puts in card : " + " " + cardRobo2.getNumber() + cardRobo2.getSuit())**;** //display card which is removed  
  
  
 } else {  
 randomRobo2 = r.nextInt(cardTrick.robo2.size())**;** //random value calculated  
 Cards cardRobo2 = cardTrick.robo2.get(randomRobo2)**;** //gets random card  
 robo2Digit = cardRobo2.getDigit()**;** //gets random digit  
 suitRobo2 = cardRobo2.getSuit()**;** //gets random suit  
 cardTrick.robo2.remove(cardRobo2)**;** //removes card  
 System.*out*.println("computer 2 puts in card : " + " " + cardRobo2.getNumber() + cardRobo2.getSuit())**;** //displays cards  
 }  
 ArrayList<Cards> robo3Arr = new ArrayList<>()**;** //creates temporary array list for robo3  
 for (int w = 0**;** w < cardTrick.robo3.size()**;** w++) {  
 if (cardTrick.robo1.get(randomRobo1).getSuit().equals(cardTrick.robo3.get(w).getSuit())) { //goes in robo3 and checks if it has a similar suit to robo1  
 robo3Arr.add(cardTrick.robo3.get(w))**;** //adds card to temporaray array list  
  
 }  
 }  
  
 if (robo3Arr.size() > 0) { //if found similar card  
 randomRobo3 = r.nextInt(robo3Arr.size())**;** //calculates random index from that list  
 Cards cardRobo3 = robo3Arr.get(randomRobo3)**;** //finds random card  
 robo3Digit = cardRobo3.getDigit()**;** //gets random card digit  
 suitRobo3 = cardRobo3.getSuit()**;** //gets random card suit  
 cardTrick.robo3.remove(cardRobo3)**;** //removes card  
 System.*out*.println("computer 3 puts in card : " + " " + cardRobo3.getNumber() + cardRobo3.getSuit())**;** //displays card  
  
  
 } else {  
 randomRobo3 = r.nextInt(cardTrick.robo3.size())**;** //random value calculated  
 Cards cardRobo3 = cardTrick.robo3.get(randomRobo3)**;** //gets random cartd using index calculated  
 robo3Digit = cardRobo3.getDigit()**;** //gets random card digit  
 suitRobo3 = cardRobo3.getSuit()**;** //gets random card suit  
 cardTrick.robo3.remove(cardRobo3)**;** //removes card  
 System.*out*.println("computer 3 puts in card : " + " " + cardRobo3.getNumber() + cardRobo3.getSuit())**;** //displays card  
 }  
 System.*out*.print("please enter your card : ")**;** userStart = input.nextLine()**;** //gets users input  
  
 int z = 0**;** for (int i = 0**;** i < cardTrick.user.size()**;** i++) {  
 if ((cardTrick.getUser().get(i).getNumber() + cardTrick.getUser().get(i).getSuit()).equals(userStart)) {  
 z += i**;** //validation done for user input  
  
  
 }  
 }  
 userDigit = cardTrick.user.get(z).getDigit()**;** //gets user entered card digit  
 suitUser = cardTrick.user.get(z).getSuit()**;** //gets user entered card suit  
  
 if ((suitRobo1.equals(suitRobo2)) && (suitRobo1.equals(suitRobo3)) && (suitRobo1.equals(suitUser))) { //checks if robo1 suit equals robo2.robo3,user  
 if ((robo1Digit > robo2Digit) && (robo1Digit > robo3Digit) && (robo1Digit > userDigit)) { //checks if robo1 digit is the highest  
 startGame = "robo1"**;** //robo1 starts game  
 System.*out*.println("the first computer wins")**;** //robo1 wins  
 robo1Points += 1**;** //robo1 points are incremented  
 } else if ((robo2Digit > robo3Digit) && (robo2Digit > userDigit) && (robo2Digit > robo1Digit)) { //checks if robo2 digit is the highest  
 startGame = "robo2"**;** //robo2 starts game  
 System.*out*.println("the second computer wins ")**;** //robo2 wins  
 robo2Points += 1**;** //robo2 points incremented  
 } else if ((robo3Digit > userDigit) && (robo3Digit > robo2Digit) && (robo3Digit > robo1Digit)) { //checks if robo3 digit is the highest  
 startGame = "robo3"**;** //robo3 starts game  
 System.*out*.println("the third computer wins ")**;** //robo3 wins  
 robo3Points += 1**;** //robo3 points incremented  
 } else {  
 startGame = "user"**;** //user starts game  
 System.*out*.println("You win")**;** //user wins  
 userPoints += 1**;** //users points are been incremented  
 }  
 } else if ((suitRobo1.equals(suitRobo2)) && (suitRobo1.equals(suitRobo3)) && (!suitUser.equals(cardTrick.trumpSuit))) { //checks if robo1 and robo2 suits are  
 if ((robo1Digit > robo2Digit) && (robo1Digit > robo3Digit)) {  
 startGame = "robo1"**;** System.*out*.println("first computer wins")**;** robo1Points += 1**;** } else if ((robo2Digit > robo1Digit) && (robo2Digit > robo3Digit)) {  
 startGame = "robo2"**;** System.*out*.println("second computer wins")**;** robo2Points += 1**;** } else {  
 startGame = "robo3"**;** System.*out*.println("third computer wins ")**;** robo3Points += 1**;** }  
 } else if ((suitRobo1.equals(suitRobo3)) && (suitRobo1.equals(suitUser)) && (!suitRobo2.equals(cardTrick.trumpSuit))) {  
 if ((robo1Digit > robo3Digit) && (robo1Digit > userDigit)) {  
 startGame = "robo1"**;** System.*out*.println("first computer wins")**;** robo1Points += 1**;** } else if ((robo3Digit > robo1Digit) && (robo3Digit > userDigit)) {  
 startGame = "robo3"**;** System.*out*.println("third computer wins")**;** robo3Points += 1**;** } else {  
 startGame = "user"**;** System.*out*.println("You win")**;** userPoints += 1**;** }  
 } else if ((suitRobo1.equals(suitRobo2)) && (suitRobo1.equals(suitUser)) && (!suitRobo3.equals(cardTrick.trumpSuit))) {  
 if ((robo1Digit > robo2Digit) && (robo1Digit > userDigit)) {  
 startGame = "robo1"**;** System.*out*.println("first computer wins")**;** robo1Points += 1**;** } else if ((robo2Digit > robo1Digit) && (robo2Digit > userDigit)) {  
 startGame = "robo2"**;** System.*out*.println("second computer wins")**;** robo2Points += 1**;** } else {  
 startGame = "user"**;** System.*out*.println("You win")**;** userPoints += 1**;** }  
 } else if ((suitRobo2.equals(cardTrick.trumpSuit)) && (suitRobo3.equals(cardTrick.trumpSuit)) && (suitUser.equals(cardTrick.trumpSuit))) {  
 if ((robo3Digit > robo2Digit) && (robo3Digit > userDigit)) {  
 startGame = "robo3"**;** System.*out*.println("the third computer wins")**;** robo3Points += 1**;** } else if ((robo2Digit > robo1Digit) && (robo2Digit > userDigit)) {  
 startGame = "robo2"**;** System.*out*.println("the second computer wins ")**;** robo2Points += 1**;** } else {  
 startGame = "user"**;** System.*out*.println("You win")**;** userPoints += 1**;** }  
 } else if ((suitRobo2.equals(cardTrick.trumpSuit)) && (suitRobo3.equals(cardTrick.trumpSuit))) {  
 if (robo2Digit > robo3Digit) {  
 startGame = "robo2"**;** System.*out*.println("the second computer wins")**;** robo2Points += 1**;** } else {  
 startGame = "robo3"**;** System.*out*.println("the third computer wins")**;** robo3Points += 1**;** }  
 } else if ((suitRobo2.equals(cardTrick.trumpSuit)) && (suitUser.equals(cardTrick.trumpSuit))) {  
 if (robo2Digit > userDigit) {  
 startGame = "robo2"**;** System.*out*.println("the second computer wins")**;** robo2Points += 1**;** } else {  
 startGame = "user"**;** System.*out*.println("You win")**;** userPoints += 1**;** }  
 } else if ((suitRobo3.equals(cardTrick.trumpSuit)) && (suitUser.equals(cardTrick.trumpSuit))) {  
 if (robo3Digit > userDigit) {  
 startGame = "robo3"**;** System.*out*.println("The third computer wins")**;** robo3Points += 1**;** } else {  
 startGame = "user"**;** System.*out*.println("You win")**;** userPoints += 1**;** }  
 } else if (suitRobo2.equals(cardTrick.trumpSuit)) {  
 startGame = "robo2"**;** System.*out*.println("the second computer wins")**;** robo2Points += 1**;** } else if (suitRobo3.equals(cardTrick.trumpSuit)) {  
 startGame = "robo3"**;** System.*out*.println("the third computer wins ")**;** robo3Points += 1**;** } else if (suitUser.equals(cardTrick.trumpSuit)) {  
 startGame = "user"**;** System.*out*.println("You win")**;** userPoints += 1**;** } else {  
 startGame = "robo1"**;** System.*out*.println("the first computer wins")**;** robo1Points += 1**;** }  
  
  
 } else if (startGame.equals("robo2")) {  
 randomRobo2 = r.nextInt(cardTrick.robo2.size())**;** Cards cardRobo2 = cardTrick.robo2.get(randomRobo2)**;** robo2Digit = cardRobo2.getDigit()**;** suitRobo2 = cardRobo2.getSuit()**;** cardTrick.robo2.remove(cardRobo2)**;** System.*out*.println("computer 2 puts card : " + " " + cardRobo2.getNumber() + cardRobo2.getSuit())**;** ArrayList<Cards> bot3temp = new ArrayList<>()**;** for (int w = 0**;** w < cardTrick.robo3.size()**;** w++) {  
 if (cardTrick.robo2.get(randomRobo2).getSuit().equals(cardTrick.robo3.get(w).getSuit())) {  
 bot3temp.add(cardTrick.robo3.get(w))**;** }  
 }  
  
 if (bot3temp.size() > 0) {  
 randomRobo3 = r.nextInt(bot3temp.size())**;** Cards cardRobo3 = bot3temp.get(randomRobo3)**;** robo3Digit = cardRobo3.getDigit()**;** suitRobo3 = cardRobo3.getSuit()**;** cardTrick.robo3.remove(cardRobo3)**;** System.*out*.println("computer 3 puts card : " + " " + cardRobo3.getNumber() + cardRobo3.getSuit())**;** } else {  
 randomRobo3 = r.nextInt(cardTrick.robo3.size())**;** Cards cardRobo3 = cardTrick.robo3.get(randomRobo3)**;** robo3Digit = cardRobo3.getDigit()**;** suitRobo3 = cardRobo3.getSuit()**;** cardTrick.robo3.remove(cardRobo3)**;** System.*out*.println("computer 3 puts in card : " + " " + cardRobo3.getNumber() + cardRobo3.getSuit())**;** }  
  
 System.*out*.print("enter a card")**;** userStart = input.nextLine()**;** int z = 0**;** for (int i = 0**;** i < cardTrick.user.size()**;** i++) {  
 if ((cardTrick.getUser().get(i).getNumber() + cardTrick.getUser().get(i).getSuit()).equals(userStart)) {  
 z += i**;** }  
 }  
 userDigit = cardTrick.user.get(z).getDigit()**;** suitUser = cardTrick.user.get(z).getSuit()**;** ArrayList<Cards> robo1Arr = new ArrayList<>()**;** for (int w = 0**;** w < cardTrick.robo1.size()**;** w++) {  
 if (cardTrick.robo2.get(randomRobo2).getSuit().equals(cardTrick.robo1.get(w).getSuit())) {  
 robo1Arr.add(cardTrick.robo1.get(w))**;** }  
 }  
  
 if (robo1Arr.size() > 0) {  
 randomRobo1 = r.nextInt(robo1Arr.size())**;** Cards cardRobo1 = robo1Arr.get(randomRobo1)**;** robo1Digit = cardRobo1.getDigit()**;** suitRobo1 = cardRobo1.getSuit()**;** cardTrick.robo1.remove(cardRobo1)**;** System.*out*.println("the first computer puts in card :" + " " + cardRobo1.getNumber() + cardRobo1.getSuit())**;** } else {  
 randomRobo1 = r.nextInt(cardTrick.robo1.size())**;** Cards cardRobo1 = cardTrick.robo1.get(randomRobo1)**;** robo1Digit = cardRobo1.getDigit()**;** suitRobo1 = cardRobo1.getSuit()**;** cardTrick.robo1.remove(cardRobo1)**;** System.*out*.println("the first computer puts in card : " + " " + cardRobo1.getNumber() + cardRobo1.getSuit())**;** }  
  
 if ((suitRobo2.equals(suitRobo3)) && (suitRobo2.equals(suitUser)) && (suitRobo2.equals(suitRobo1))) {  
 if ((robo2Digit > robo1Digit) && (robo2Digit > robo3Digit) && (robo2Digit > userDigit)) {  
 startGame = "robo2"**;** System.*out*.println("the second computer wins")**;** robo2Points += 1**;** } else if ((robo3Digit > robo1Digit) && (robo3Digit > robo2Digit) && (robo3Digit > userDigit)) {  
 startGame = "robo3"**;** System.*out*.println("the third computer wins")**;** robo3Points += 1**;** } else if ((robo1Digit > robo2Digit) && (robo1Digit > robo3Digit) && (robo1Digit > userDigit)) {  
 startGame = "robo1"**;** System.*out*.println("the first computer wins")**;** robo1Points += 1**;** } else {  
 startGame = "user"**;** System.*out*.println("You win")**;** userPoints += 1**;** }  
 } else if ((suitRobo2.equals(suitRobo3)) && (suitRobo2.equals(suitUser)) && (!suitRobo1.equals(cardTrick.trumpSuit))) {  
 if ((robo2Digit > robo3Digit) && (robo2Digit > userDigit)) {  
 startGame = "robo2"**;** System.*out*.println("the second computer wins ")**;** robo2Points += 1**;** } else if ((robo3Digit > robo2Digit) && (robo3Digit > userDigit)) {  
 startGame = "robo3"**;** System.*out*.println("the third computer wins")**;** robo3Points += 1**;** } else {  
 startGame = "user"**;** System.*out*.println("You win")**;** userPoints += 1**;** }  
 } else if ((suitRobo2.equals(suitUser)) && (suitRobo2.equals(suitRobo1)) && (!suitRobo3.equals(cardTrick.trumpSuit))) {  
 if ((robo2Digit > userDigit) && (robo2Digit > robo1Digit)) {  
 startGame = "robo2"**;** System.*out*.println("the second computer wins")**;** robo2Points += 1**;** } else if ((userDigit > robo1Digit) && (userDigit > robo2Digit)) {  
 startGame = "user"**;** System.*out*.println("You win")**;** userPoints += 1**;** } else {  
 startGame = "robo1"**;** System.*out*.println("the first computer wins")**;** robo1Points += 1**;** }  
 } else if ((suitRobo2.equals(suitRobo3)) && (suitRobo2.equals(suitRobo1)) && (!suitUser.equals(cardTrick.trumpSuit))) {  
 if ((robo2Digit > robo3Digit) && (robo2Digit > robo1Digit)) {  
 startGame = "robo2"**;** System.*out*.println("the second computer wins")**;** robo2Points += 1**;** } else if ((robo1Digit > robo2Digit) && (robo1Digit > robo3Digit)) {  
 startGame = "robo1"**;** System.*out*.println("the first computer wins")**;** robo1Points += 1**;** } else {  
 startGame = "robo3"**;** System.*out*.println("the third computer wins")**;** robo3Points += 1**;** }  
 } else if ((suitRobo3.equals(cardTrick.trumpSuit)) && (suitUser.equals(cardTrick.trumpSuit)) && (suitRobo1.equals(cardTrick.trumpSuit))) {  
 if ((robo3Digit > userDigit) && (robo3Digit > robo1Digit)) {  
 startGame = "robo3"**;** System.*out*.println("the third computer wins")**;** robo3Points += 1**;** } else if ((robo1Digit > userDigit) && (robo1Digit > robo3Digit)) {  
 startGame = "robo1"**;** System.*out*.println("the first computer wins ")**;** robo1Points += 1**;** } else {  
 startGame = "user"**;** System.*out*.println("You win")**;** userPoints += 1**;** }  
 } else if ((suitRobo3.equals(cardTrick.trumpSuit)) && (suitUser.equals(cardTrick.trumpSuit))) {  
 if ((robo3Digit > userDigit)) {  
 startGame = "robo3"**;** System.*out*.println("the third computer wins")**;** robo3Points += 1**;** } else {  
 startGame = "user"**;** System.*out*.println("You win")**;** userPoints += 1**;** }  
 } else if ((suitRobo3.equals(cardTrick.trumpSuit)) && (suitRobo1.equals(cardTrick.trumpSuit))) {  
 if (robo3Digit > robo1Digit) {  
 startGame = "robo3"**;** System.*out*.println("The third computer wins ")**;** robo3Points += 1**;** } else {  
 startGame = "robo1"**;** System.*out*.println("The first computer wins")**;** robo1Points += 1**;** }  
 } else if ((suitUser.equals(cardTrick.trumpSuit)) && (suitRobo1.equals(cardTrick.trumpSuit))) {  
 if (userDigit > robo1Digit) {  
 startGame = "user"**;** System.*out*.println("You win")**;** userPoints += 1**;** } else {  
 startGame = "robo1"**;** System.*out*.println("The first computer wins")**;** robo1Points += 1**;** }  
 } else if (suitRobo3.equals(cardTrick.trumpSuit)) {  
 startGame = "robo3"**;** System.*out*.println("the third computer wins ")**;** robo3Points += 1**;** } else if (suitUser.equals(cardTrick.trumpSuit)) {  
 startGame = "user"**;** System.*out*.println("You win")**;** userPoints += 1**;** } else if (suitRobo1.equals(cardTrick.trumpSuit)) {  
 startGame = "robo1"**;** System.*out*.println("the first computer wins ")**;** robo1Points += 1**;** } else {  
 startGame = "robo2"**;** System.*out*.println("the second computer wins")**;** robo2Points += 1**;** }  
  
 for (int x = 0**;** x <= cardTrick.getUser().size() - 1**;** x++) {  
 System.*out*.println("Your cards are : " + cardTrick.getUser().get(x).getNumber() + cardTrick.getUser().get(x).getSuit())**;** //prints users cards  
 }  
 } else {  
 randomRobo3 = r.nextInt(cardTrick.robo3.size())**;** //takes a random index  
 Cards cardRobo3 = cardTrick.robo3.get(randomRobo3)**;** //assigns random index card to cardrobo3  
 robo3Digit = cardRobo3.getDigit()**;** //get digit  
 suitRobo3 = cardRobo3.getSuit()**;** //get suit  
 cardTrick.robo3.remove(cardRobo3)**;** //removes card  
 System.*out*.println("the third computer puts in card : " + " " + cardRobo3.getNumber() + cardRobo3.getSuit())**;** //prints cards  
  
 System.*out*.print("please enter your card")**;** //enter users card  
 userStart = input.nextLine()**;** //takes users input from user  
  
 int z = 0**;** for (int i = 0**;** i < cardTrick.user.size()**;** i++) {  
 if ((cardTrick.getUser().get(i).getNumber() + cardTrick.getUser().get(i).getSuit()).equals(userStart)) {  
 z += i**;** //checks if user has entered a valid input  
  
 }  
 }  
 userDigit = cardTrick.user.get(z).getDigit()**;** //gets users digit  
 suitUser = cardTrick.user.get(z).getSuit()**;** //gets users suit  
  
 ArrayList<Cards> robo1Arr = new ArrayList<>()**;** //temporary array list created  
 for (int w = 0**;** w < cardTrick.robo3.size()**;** w++) {  
 if (cardTrick.robo3.get(randomRobo3).getSuit().equals(cardTrick.robo1.get(w).getSuit())) { //checks if a card in robo3 deck has a suit equal to robo1  
 robo1Arr.add(cardTrick.robo1.get(w))**;** //if has adds card to temporaray array list  
  
  
 }  
 }  
 if (robo1Arr.size() > 0) { //if similar suit cards found  
 randomRobo1 = r.nextInt(robo1Arr.size())**;** //gets a random index from array  
 Cards cardRobo1 = robo1Arr.get(randomRobo1)**;** //gets random card  
 robo1Digit = cardRobo1.getDigit()**;** //gets digit  
 suitRobo1 = cardRobo1.getSuit()**;** //get suit  
 cardTrick.robo1.remove(cardRobo1)**;** //removes random card  
 System.*out*.println("the first computer puts in card : " + " " + cardRobo1.getNumber() + cardRobo1.getSuit())**;** //prints card  
  
  
 } else {  
 randomRobo1 = r.nextInt(cardTrick.robo1.size())**;** //gets a random index from array  
 Cards cardRobo1 = cardTrick.robo1.get(randomRobo1)**;** //stores random card calculated from index at temporaray arrayh  
 robo1Digit = cardRobo1.getDigit()**;** //gets digit  
 suitRobo1 = cardRobo1.getSuit()**;** //get suit  
 cardTrick.robo1.remove(cardRobo1)**;** //removes card  
 System.*out*.println("the first computer puts in card : " + " " + cardRobo1.getNumber() + cardRobo1.getSuit())**;** //prints card  
 }  
  
 ArrayList<Cards> robo2Arr = new ArrayList<>()**;** //creates a temporary array for robo2  
 for (int w = 0**;** w < cardTrick.robo2.size()**;** w++) {  
 if (cardTrick.robo3.get(randomRobo3).getSuit().equals(cardTrick.robo2.get(w).getSuit())) { //goes inside robo2 deck checks if there is a card suit similar to robo2  
 robo2Arr.add(cardTrick.robo2.get(w))**;** }  
 }  
  
 if (robo2Arr.size() > 0) {  
 randomRobo2 = r.nextInt(robo2Arr.size())**;** Cards cardRobo2 = robo2Arr.get(randomRobo2)**;** robo2Digit = cardRobo2.getDigit()**;** suitRobo2 = cardRobo2.getSuit()**;** cardTrick.robo2.remove(cardRobo2)**;** System.*out*.println("the second computer puts in card : " + " " + cardRobo2.getNumber() + cardRobo2.getSuit())**;** } else {  
 randomRobo2 = r.nextInt(cardTrick.robo2.size())**;** Cards cardRobo2 = cardTrick.robo2.get(randomRobo2)**;** robo2Digit = cardRobo2.getDigit()**;** suitRobo2 = cardRobo2.getSuit()**;** cardTrick.robo2.remove(cardRobo2)**;** System.*out*.println("the second computer puts in card : " + " " + cardRobo2.getNumber() + cardRobo2.getSuit())**;** }  
  
 if ((suitRobo3.equals(suitUser)) && (suitRobo3.equals(suitRobo1)) && (suitRobo3.equals(suitRobo2))) {  
 if ((robo3Digit > userDigit) && (robo3Digit > robo1Digit) && (robo3Digit > robo2Digit)) {  
 startGame = "robo3"**;** System.*out*.println("the third computer wins")**;** robo3Points += 1**;** } else if ((robo1Digit > robo3Digit) && (robo1Digit > robo2Digit) && (robo1Digit > userDigit)) {  
 startGame = "robo1"**;** System.*out*.println("the first computer wins")**;** robo1Points += 1**;** } else if ((robo2Digit > robo3Digit) && (robo2Digit > robo1Digit) && (robo2Digit > userDigit)) {  
 startGame = "robo2"**;** System.*out*.println("the second computer wins")**;** robo2Points += 1**;** } else {  
 startGame = "user"**;** System.*out*.println("You win ")**;** userPoints += 1**;** }  
 } else if (suitRobo3.equals(suitUser) && (suitRobo3.equals(suitRobo1)) && (!suitRobo2.equals(cardTrick.trumpSuit))) {  
 if ((robo3Digit > userDigit) && (robo3Digit > robo1Digit)) {  
 startGame = "robo3"**;** System.*out*.println("the third computer wins ")**;** robo3Points += 1**;** } else if ((robo1Digit > robo3Digit) && (robo1Digit > userDigit)) {  
 startGame = "robo1"**;** System.*out*.println("the first computer wins ")**;** robo1Points += 1**;** } else {  
 startGame = "user"**;** System.*out*.println("You win")**;** userPoints += 1**;** }  
 } else if ((suitRobo3.equals(suitRobo1)) && (suitRobo3.equals(suitRobo2)) && (!suitUser.equals(cardTrick.trumpSuit))) {  
 if ((robo3Digit > robo1Digit) && (robo3Digit > robo2Digit)) {  
 startGame = "robo3"**;** System.*out*.println("the third computer wins")**;** robo3Points += 1**;** } else if ((robo2Digit > robo1Digit) && (robo2Digit > robo3Digit)) {  
 startGame = "robo2"**;** System.*out*.println("the second computer wins")**;** robo2Points += 1**;** } else {  
 startGame = "robo1"**;** System.*out*.println("the first computer wins")**;** robo1Points += 1**;** }  
 } else if ((suitRobo3.equals(suitUser)) && (suitRobo3.equals(suitRobo2)) && (!suitRobo1.equals(cardTrick.trumpSuit))) {  
 if ((robo3Digit > userDigit) && (robo3Digit > robo2Digit)) {  
 startGame = "robo3"**;** System.*out*.println("the third computer wins")**;** robo3Points += 1**;** } else if ((robo2Digit > userDigit) && (robo2Digit > robo3Digit)) {  
 startGame = "robo2"**;** System.*out*.println("the second computer wins")**;** robo2Points += 1**;** } else {  
 startGame = "user"**;** System.*out*.println("You win")**;** userPoints += 1**;** }  
 } else if ((suitUser.equals(cardTrick.trumpSuit)) && (suitRobo1.equals(cardTrick.trumpSuit)) && (suitRobo2.equals(cardTrick.trumpSuit))) {  
 if ((userDigit > robo1Digit) && (userDigit > robo2Digit)) {  
 startGame = "user"**;** System.*out*.println("You win ")**;** userPoints += 1**;** } else if ((robo1Digit > userDigit) && (robo1Digit > robo2Digit)) {  
 startGame = "robo1"**;** System.*out*.println("the first computer wins ")**;** robo1Points += 1**;** } else {  
 startGame = "robo2"**;** System.*out*.println("the second computer wins ")**;** robo2Points += 1**;** }  
 } else if ((suitUser.equals(cardTrick.trumpSuit)) && (suitRobo1.equals(cardTrick.trumpSuit))) {  
 if ((userDigit > robo1Digit)) {  
 startGame = "user"**;** System.*out*.println("You win ")**;** userPoints += 1**;** } else {  
 startGame = "robo1"**;** System.*out*.println("the first computer wins")**;** robo1Points += 1**;** }  
 } else if ((suitUser.equals(cardTrick.trumpSuit)) && (suitRobo2.equals(cardTrick.trumpSuit))) {  
 if (userDigit > robo2Digit) {  
 startGame = "user"**;** System.*out*.println("You win")**;** userPoints += 1**;** } else {  
 startGame = "robo2"**;** System.*out*.println("the second computer wins ")**;** robo2Points += 1**;** }  
 } else if ((suitRobo1.equals(cardTrick.trumpSuit)) && (suitRobo2.equals(cardTrick.trumpSuit))) {  
 if (robo1Digit > robo2Digit) {  
 startGame = "bot1"**;** System.*out*.println("the first computer wins ")**;** robo1Points += 1**;** } else {  
 startGame = "robo2"**;** System.*out*.println("the second computer wins")**;** robo2Points += 1**;** }  
 } else if (suitUser.equals(cardTrick.trumpSuit)) {  
 startGame = "user"**;** System.*out*.println("You win")**;** userPoints += 1**;** } else if (suitRobo1.equals(cardTrick.trumpSuit)) {  
 startGame = "robo1"**;** System.*out*.println("the first computer wins")**;** robo1Points += 1**;** } else if (suitRobo2.equals(cardTrick.trumpSuit)) {  
 startGame = "robo2"**;** System.*out*.println("the second computer wins")**;** robo2Points += 1**;** } else {  
 startGame = "robo3"**;** System.*out*.println("the third computer wins")**;** robo3Points += 1**;** }  
 for (int x = 0**;** x <= cardTrick.getUser().size() - 1**;** x++) {  
 System.*out*.println("Your Card is : " + cardTrick.getUser().get(x).getNumber() + cardTrick.getUser().get(x).getSuit())**;** }  
  
 }  
 }  
 System.*out*.println("Your score : "+ userPoints)**;** System.*out*.println("Computer 1 score : "+ robo1Points)**;** System.*out*.println("Computer 2 score : "+ robo2Points)**;** System.*out*.println("Computer 3 score : "+ robo3Points)**;** if (leadTrump.equals("user")) {  
 leadTrump = "robo1"**;** } else if (leadTrump.equals("robo1")) {  
 leadTrump = "robo2"**;** } else if (leadTrump.equals("robo2")) {  
 leadTrump = "robo3"**;** } else {  
 leadTrump = "user "**;** }  
  
 System.*out*.println("This Game Is Over .... Do you wish to play again ? (click on 1 to play again and 2 to stop ")**;** playagain = input.nextInt()**;** }  
 }  
}

public class CardDeck {  
  
 Cards card1 = new Cards("♦"**,** "7"**,** 7)**;** //card 1 stores 7 0f diamonds  
 Cards card2 = new Cards("♦"**,** "8"**,** 8)**;** //card 2 stores 8 0f diamonds  
 Cards card3 = new Cards("♦"**,** "9"**,** 9)**;** //card 3 stores 9 0f diamonds  
 Cards card4 = new Cards("♦"**,** "10"**,** 10)**;** //card 4 stores 10 0f diamonds  
 Cards card5 = new Cards("♦"**,** "J"**,** 11)**;** //card 5 stores jack 0f diamonds  
 Cards card6 = new Cards("♦"**,** "Q"**,** 12)**;** //card 6 stores queen 0f diamonds  
 Cards card7= new Cards("♦"**,** "K"**,** 13)**;** //card 7 stores king 0f diamonds  
 Cards card8 = new Cards("♦"**,** "A"**,** 14)**;** //card 8 stores ace of diamonds  
  
 Cards card9 = new Cards("♥"**,** "7"**,** 7)**;** //card9 stores the 7 of hearts  
 Cards card10 = new Cards("♥"**,** "8"**,** 8)**;** //card10 stores the 8 of hearts  
 Cards card11 = new Cards("♥"**,** "9"**,** 9)**;** //card11 stores the 9 of hearts  
 Cards card12 = new Cards("♥"**,** "10"**,** 10)**;** //card12 stores the 10 of hearts  
 Cards card13 = new Cards("♥"**,** "J"**,** 11)**;** //card13 stores the jack of hearts  
 Cards card14 = new Cards("♥"**,** "Q"**,** 12)**;** //card14 stores the queen of hearts  
 Cards card15 = new Cards("♥"**,** "K"**,** 13)**;** //card15 stores the king of hearts  
 Cards card16 = new Cards("♥"**,** "A"**,** 14)**;** //card16 stores the ace of hearts  
  
  
 Cards card17 = new Cards ("♠"**,** "7"**,** 7)**;** //card17 stores the 7 of spades  
 Cards card18 = new Cards("♠"**,** "8"**,** 8)**;** //card18 stores the 8 of spades  
 Cards card19 = new Cards("♠"**,** "9"**,** 9)**;** //card 19 stores the 9 of spades  
 Cards card20 = new Cards("♠"**,** "10"**,** 10)**;** //card20 stores the 10 of spades  
 Cards card21 = new Cards("♠"**,** "J"**,** 11)**;** //card21 stores the jack of spades  
 Cards card22 = new Cards("♠"**,** "Q"**,** 12)**;** //card22 stores the queen of spades  
 Cards card23 = new Cards("♠"**,** "K"**,** 13)**;** //card23 stores the king of spades  
 Cards card24 = new Cards("♠"**,** "A"**,** 14)**;** //card24 stores the ace of spades  
  
  
  
 Cards card25 = new Cards("♣"**,** "7"**,** 7)**;** //card 25 stores the 7 of clubs  
 Cards card26 = new Cards("♣"**,** "8"**,** 8)**;** //card 26 stores the 8 of clubs  
 Cards card27 = new Cards("♣"**,** "9"**,** 9)**;** //card 27 stores the 9 of clubs  
 Cards card28 = new Cards("♣"**,** "10"**,** 10)**;** //card 28 stores the 10 of clubs  
 Cards card29 = new Cards("♣"**,** "J"**,** 11)**;** //card29 stores the jack of clubs  
 Cards card30 = new Cards("♣"**,** "Q"**,** 12)**;** //card30 stores the queen of clubs  
 Cards card31 = new Cards("♣"**,** "K"**,** 13)**;** //card31 stores the king of clubs  
 Cards card32 = new Cards("♣"**,** "A"**,** 14)**;** //cad32 stores the ace of clubs  
  
  
  
  
  
 public Cards CheckCard(String UserInput) {  
 Cards userStartPlay = null**;** //input given by user (userStartPlay) initially set to null  
 if (UserInput.equals("7♦")) //checks if users input is equal to 7 of diamonds  
 userStartPlay = card1**;** //assigns to card1  
 else if (UserInput.equals("8♦")) //checks if users input is equal to 8 of diamonds  
 userStartPlay = card2**;** //assigns to card2  
 else if (UserInput.equals("9♦")) //checks if users input is equal to 9 of diamonds  
 userStartPlay = card3**;** //assigns to card3  
 else if (UserInput.equals("10♦")) //checks if users input is equal to 10 of diamonds  
 userStartPlay = card4**;** //assigns to card4  
 else if (UserInput.equals("J♦")) //checks if users input is equal to jack of diamonds  
 userStartPlay = card5**;** //assigns to card5  
 else if (UserInput.equals("Q♦")) //checks if users input is equal to queen of diamonds  
 userStartPlay = card6**;** //assigns to card6  
 else if (UserInput.equals("K♦")) //checks if users input is equal to king of diamonds  
 userStartPlay = card7**;** //assigns to card7  
 else if (UserInput.equals("A♦")) //checks if users input is equal to ace of diamonds  
 userStartPlay = card8**;** //assigns to card8  
 else if (UserInput.equals("7♥")) //checks if users input is equal to 7 of hearts  
 userStartPlay = card9**;** //assigns to card9  
 else if (UserInput.equals("8♥")) //checks if users input is equal to 8 of hearts  
 userStartPlay = card10**;** //assigns to card10  
 else if (UserInput.equals("9♥")) //checks if users input is equal to 9 of hearts  
 userStartPlay = card11**;** //assigns to card11  
 else if (UserInput.equals("10♥")) //checks if users input is equal to 10 of hearts  
 userStartPlay = card12**;** //assigns to card12  
 else if (UserInput.equals("J♥")) //checks if users input is equal to jack of hearts  
 userStartPlay = card13**;** //assigns to card13  
 else if (UserInput.equals("Q♥")) //checks if users input is equal to queen of hearts  
 userStartPlay = card14**;** //assigns to card14  
 else if (UserInput.equals("K♥")) //checks if users input is equal to king of hearts  
 userStartPlay = card15**;** //assigns to card15  
 else if (UserInput.equals("A♥")) //checks if users input is equal to ace of hearts  
 userStartPlay = card16**;** //assigns to card16  
 else if (UserInput.equals("7♠")) //checks if users input is equal to 7 of spades  
 userStartPlay = card17**;** //assigns to card17  
 else if (UserInput.equals("8♠")) //checks if users input is equal to 8 of spades  
 userStartPlay = card18**;** //assigns to card18  
 else if (UserInput.equals("9♠")) //checks if users input is equal to 9 of spades  
 userStartPlay = card19**;** //assigns to card19  
 else if (UserInput.equals("10♠")) //checks if users input is equal to 10 of spades  
 userStartPlay = card20**;** //assigns to card20  
 else if (UserInput.equals("J♠")) //checks if users input is equal to jack of spades  
 userStartPlay = card21**;** //assigns to card21  
 else if (UserInput.equals("Q♠")) //checks if users input is equal to queen of spades  
 userStartPlay = card22**;** //assigns to card22  
 else if (UserInput.equals("K♠")) //checks if users input is equal to king of spades  
 userStartPlay = card23**;** //assigns to card23  
 else if (UserInput.equals("A♠")) //checks if users input is equal to ace of spades  
 userStartPlay = card24**;** //assigns to card24  
 else if (UserInput.equals("7♣")) //checks if user input is equal to 7 of clubs  
 userStartPlay = card25**;** //assigns to card25  
 else if (UserInput.equals("8♣")) //checks if user input is equal to 8 of clubs  
 userStartPlay = card26**;** //assigns to card26  
 else if (UserInput.equals("9♣")) //checks if user input is equal to 9 of clubs  
 userStartPlay = card27**;** //assigns to card27  
 else if (UserInput.equals("10♣")) //checks if user input is equal to 10 of clubs  
 userStartPlay = card28**;** //assigns to card28  
 else if (UserInput.equals("J♣")) //checks if user input is equal to jack of clubs  
 userStartPlay = card29**;** //assigns to card29  
 else if (UserInput.equals("Q♣")) //checks if user input is equal to queen of clubs  
 userStartPlay = card30**;** //assigns to card30  
 else if (UserInput.equals("K♣")) //checks if user input is equal to king of clubs  
 userStartPlay = card31**;** //assigns to card31  
 else if (UserInput.equals("A♣")) //checks if user input is equal to ace of clubs  
 userStartPlay = card32**;** //assigns to card32  
 return userStartPlay**;** //returns users input  
 }  
  
  
  
}

public class Cards {  
 private String suit**;** //stores the suit of the card  
 private String number**;** //stores the card value  
 private int digit**;** //stores the digit which the card represents  
  
 public Cards(String suit**,** String number**,** int digit){ //this is the constructor for the class cards  
 this.suit = suit**;** //initialises the suit  
 this.number = number**;** //initialises the number  
 this.digit = digit**;** //initialises the digit  
 }  
  
 public String getSuit() { return this.suit**;**} //returns the card suit  
 public String getNumber() { return this.number**;**} //returns card number  
 public int getDigit() { return this.digit**;**} //returns card digit  
}

import java.util.ArrayList**;**import java.util.Scanner**;**import java.util.Random**;**public class Skill {  
 Scanner sc = new Scanner(System.*in*)**;** public ArrayList<Cards> cardDeck = new ArrayList<>()**;** //carddeck array list created  
 public ArrayList<Cards> user = new ArrayList<>()**;** //user array list created  
 public ArrayList<Cards> robo1 = new ArrayList<>()**;** //robo1 array list created  
 public ArrayList<Cards> robo2 = new ArrayList<>()**;** //robo2 array list created  
 public ArrayList<Cards> robo3 = new ArrayList<>()**;** //robo3 array list created  
 Random r = new Random()**;** public void addCards(Cards... currentCard) { //goes in a loop  
 System.*out*.println(currentCard.length)**;** for (int i = 0**;** i < currentCard.length **;** i++) {  
 cardDeck.add(currentCard[i])**;** //adds current card to card deck  
 System.*out*.println(i)**;** }  
 System.*out*.println(cardDeck)**;** }  
  
 public ArrayList<Cards> getCardDeck() {  
 return cardDeck**;** } //gets carddeck  
  
 public void shuffleDeck(ArrayList<Cards> shuffleArr) { //shuffling algrorithm implemented  
 int element = r.nextInt(32)**;** //gets a random number from 0-31  
 for (int i = 0**;** i <= shuffleArr.size()+1 **;** i++) { //goes in a loop  
  
 Cards newCard = shuffleArr.get(element)**;** //swaps random elememts card with card in the index  
 shuffleArr.remove(element)**;** shuffleArr.add(0**,** newCard)**;** //adds to a shuffling array  
 }  
  
 }  
  
 public void leadUser() { //if user leads the trumps  
 for (int i = 0**;** i < 3**;** i++) { //first 4 cards user  
 Cards card = cardDeck.get(i)**;** user.add(card)**;** //addded to user array  
 }  
 for (int i = 4**;** i < 8**;** i++) { //next 4 cards to robo1  
 Cards card = cardDeck.get(i)**;** robo1.add(card)**;** //added to robo1 array  
 }  
 for (int i = 8**;** i < 12**;** i++) { //next 4 cards to robo2  
 Cards card = cardDeck.get(i)**;** robo2.add(card)**;** //added to robo2 array  
 }  
 for (int i = 12**;** i < 16**;** i++) { //next 4 cards to robo3  
 Cards card = cardDeck.get(i)**;** robo3.add(card)**;** //added to robo3 array  
 }  
 for (int i = 16**;** i < 20**;** i++) { //next 4 cards to user  
 Cards card = cardDeck.get(i)**;** user.add(card)**;** //addded to user array  
 }  
 for (int i = 20**;** i < 24**;** i++) { //next 4 cards to robo1  
 Cards card = cardDeck.get(i)**;** robo1.add(card)**;** //added to robo1 array  
 }  
 for (int i = 24**;** i < 28**;** i++) { //next 4 cards to robo2  
 Cards card = cardDeck.get(i)**;** robo2.add(card)**;** //added to robo2 array  
 }  
 for (int i = 28**;** i < 31**;** i++) { //next 4 cards to robo3  
 Cards card = cardDeck.get(i)**;** robo3.add(card)**;** //added to robo3 array  
 }  
  
 }  
  
 public void leadRobo1() { //if robo1 leads the trumps  
 for (int i = 0**;** i < 4**;** i++) { //first four cards given to robo1  
 Cards card = cardDeck.get(i)**;** robo1.add(card)**;** //added to robo1 array  
 }  
 for (int i = 4**;** i < 8**;** i++) { //next 4 cards to robo2  
 Cards card = cardDeck.get(i)**;** robo2.add(card)**;** //added to robo2 array  
 }  
 for (int i = 8**;** i < 12**;** i++) { //next 4 cards to robo3  
 Cards card = cardDeck.get(i)**;** robo3.add(card)**;** //added to robo3 array  
 }  
 for (int i = 12**;** i < 16**;** i++) { //next 4 cards to user  
 Cards card = cardDeck.get(i)**;** user.add(card)**;** //addded to user array  
 }  
 for (int i = 16**;** i < 20**;** i++) { //next 4 cards to robo1  
 Cards card = cardDeck.get(i)**;** robo1.add(card)**;** //added to robo1 array  
 }  
 for (int i = 20**;** i < 24**;** i++) { //next 4 cards to robo2  
 Cards card = cardDeck.get(i)**;** robo2.add(card)**;** //added to robo2 array  
 }  
 for (int i = 24**;** i < 28**;** i++) { //next 4 cards to robo3  
 Cards card = cardDeck.get(i)**;** robo3.add(card)**;** //added to robo3 array  
 }  
 for (int i = 28**;** i < 32**;** i++) { //next 4 cards to user  
 Cards card = cardDeck.get(i)**;** user.add(card)**;** //added to user array  
 }  
  
 }  
  
 public void leadRobo2() { //if robo2 leads trumps  
 for (int i = 0**;** i < 4**;** i++) { //first four cards to robo2  
 Cards card = cardDeck.get(i)**;** robo2.add(card)**;** //added to robo2 array  
 }  
 for (int i = 4**;** i < 8**;** i++) { //next 4 cards to robo3  
 Cards card = cardDeck.get(i)**;** robo3.add(card)**;** //added to robo3 array  
 }  
 for (int i = 8**;** i < 12**;** i++) { //next 4 cards to user  
 Cards card = cardDeck.get(i)**;** user.add(card)**;** //added to user array  
 }  
 for (int i = 12**;** i < 16**;** i++) { //next 4 cards to robo1  
 Cards card = cardDeck.get(i)**;** robo1.add(card)**;** //added to robo1 array  
 }  
 for (int i = 16**;** i < 20**;** i++) { //next 4 cards to robo2  
 Cards card = cardDeck.get(i)**;** robo2.add(card)**;** //added to robo2 array  
 }  
 for (int i = 20**;** i < 24**;** i++) { //next 4 cards to robo3  
 Cards card = cardDeck.get(i)**;** robo3.add(card)**;** //added to robo3 array  
 }  
 for (int i = 24**;** i < 28**;** i++) { //next 4 cards to user  
 Cards card = cardDeck.get(i)**;** user.add(card)**;** //added to user array  
 }  
 for (int i = 28**;** i < 32**;** i++) { //next 4 cards to robo1  
 Cards card = cardDeck.get(i)**;** robo1.add(card)**;** //added to robo1 array  
 }  
  
 }  
  
 public void leadRobo3() { //if robo3 leads trumps  
 for (int i = 0**;** i < 4**;** i++) { //first four cards given to robo3  
 Cards card = cardDeck.get(i)**;** robo3.add(card)**;** //added to robo3 array  
 }  
 for (int i = 4**;** i < 8**;** i++) { //next 4 cards to user  
 Cards card = cardDeck.get(i)**;** user.add(card)**;** //added to user array  
 }  
 for (int i = 8**;** i < 12**;** i++) { //next 4 cards to robo1  
 Cards card = cardDeck.get(i)**;** robo1.add(card)**;** //added to robo1 array  
 }  
 for (int i = 12**;** i < 16**;** i++) { //next 4 cards to robo2  
 Cards card = cardDeck.get(i)**;** robo2.add(card)**;** //added to robo2 array  
 }  
 for (int i = 16**;** i < 20**;** i++) { //next 4 cards to robo3  
 Cards card = cardDeck.get(i)**;** robo3.add(card)**;** //added to robo3 array  
 }  
 for (int i = 20**;** i < 24**;** i++) { //next 4 cards to user  
 Cards card = cardDeck.get(i)**;** user.add(card)**;** //added to user array  
 }  
 for (int i = 24**;** i < 28**;** i++) { //next 4 cards to robo1  
 Cards card = cardDeck.get(i)**;** robo1.add(card)**;** //added to robo1 array  
 }  
 for (int i = 28**;** i < 32**;** i++) { //next 4 cards to robo2  
 Cards card = cardDeck.get(i)**;** robo2.add(card)**;** //added to robo2 array  
 }  
  
 }  
  
 public ArrayList<Cards> getRobo1() {  
 return robo1**;** } //returns robo1 array  
  
 public ArrayList<Cards> getUser() {  
 return user**;** } //returns user array  
  
 public ArrayList<Cards> getRobo2() {  
 return robo2**;** } //returns robo2 array  
 public ArrayList<Cards> getRobo3() {  
 return robo3**;** } //returns robo3 array  
  
String trumpSuit =null**;** //initialises trump suit as null  
 public void trumpUser(ArrayList<Cards> arr){  
 String userTrump**;** //gets users trump  
 leadUser()**;** //divides cards equally starting from user  
  
 for(int x=0**;** x<=3**;** x++) {  
 System.*out*.println("Your cards are : "+ getUser().get(x).getNumber()+ getUser().get(x).getSuit())**;** //displays users cards  
 }  
 boolean flag = false**;** //flag assigned to false  
 while (!flag){ //while flag not equal to false  
 flag = true**;** //flag assigned to true  
 System.*out*.println("please select a trump suit from : diamonds ,hearts ,spades ,clubs ")**;** userTrump = sc.nextLine()**;** //gets users trump  
 switch (userTrump) {  
 case "diamonds": //checks if user has put diamonds as trumps  
 trumpSuit = "♦"**;** //trump made diamonds  
 break**;** case "hearts": //checks if user has put hearts as trumps  
 trumpSuit = "♥"**;** //trump made hearts  
 break**;** case "spades": //checks if user has put spades as trumps  
 trumpSuit = "♠"**;** //trumps made spades  
 break**;** case "clubs": //checks if user has put clubs as trumps  
 trumpSuit = "♣"**;** //trumps made clubs  
 break**;** default:  
 flag = false**;** //flag assigned to false  
 }  
 }  
  
 for(int x = 0**;** x<= getUser().size()-1**;** x++) {  
 System.*out*.println("your cards are : " + getUser().get(x).getNumber() + getUser().get(x).getSuit())**;** //prints users cards  
 }  
  
 System.*out*.println("the trump suit is : "+ trumpSuit)**;** //prints trump for that round  
  
 }  
  
 //trump selection for robo  
 public void roboChoiceTrump(ArrayList<Cards> cardArr) {  
 int RandomIndex = r.nextInt(4)**;** //gets a random index  
 //checks if all the four cards have same suit  
 if ((cardArr.get(0).getSuit().equals(cardArr.get(1).getSuit())) == (cardArr.get(2).getSuit().equals(cardArr.get(3).getSuit()))){  
 trumpSuit = cardArr.get(0).getSuit()**;** //assigms trumps to one of the card since all hve same suit  
 System.*out*.println("the trump suit is : "+ trumpSuit)**;** //prints trumps  
  
 }  
 //checks if three card suits are equal  
 //checks if first , second and third catrd suits the same  
 else if((cardArr.get(0).getSuit().equals(cardArr.get(1).getSuit())) && (cardArr.get(1).getSuit().equals(cardArr.get(2).getSuit()))){  
 trumpSuit = cardArr.get(0).getSuit()**;** //chooses one of the cards  
 System.*out*.println("the trump suit is : "+ trumpSuit)**;** //prints trumps  
 }  
  
  
 else if((cardArr.get(0).getSuit().equals(cardArr.get(2).getSuit())) && (cardArr.get(2).getSuit().equals(cardArr.get(3).getSuit()))){ //checks if first ,third and fourth suit same  
 trumpSuit = cardArr.get(0).getSuit()**;** //chooses one of the cards with same suit  
 System.*out*.println("the trump suit is : "+ trumpSuit)**;** //prints trumps  
 }  
 else if((cardArr.get(1).getSuit().equals(cardArr.get(2).getSuit())) && (cardArr.get(2).getSuit().equals(cardArr.get(3).getSuit()))){ //checks if second ,third and fouth card suit same  
 trumpSuit = cardArr.get(0).getSuit()**;** //chooses one of the cards with same suit  
 System.*out*.println("the trump suit is : "+ trumpSuit)**;** //prints trumps  
 }  
 else if((cardArr.get(0).getSuit().equals(cardArr.get(1).getSuit())) && (cardArr.get(1).getSuit().equals(cardArr.get(3).getSuit()))){ //checks if first,second and third card suit same  
 trumpSuit = cardArr.get(0).getSuit()**;** //chooses one card with same suit  
 System.*out*.println("the trump suit is : "+ trumpSuit)**;** //prints trumps  
 }  
  
 //checks if there are two similar card suits  
 else if((cardArr.get(0).getSuit().equals(cardArr.get(1).getSuit())) && (cardArr.get(2).getSuit().equals(cardArr.get(3).getSuit()))){ //checks if first and second equal and third and fourth equal  
  
 if(cardArr.get(0).getDigit()==14 || cardArr.get(1).getDigit()==14){ //checks if either first or second card has an ace  
 trumpSuit = cardArr.get(2).getSuit()**;** //if ace present gives to third and fourth card suit  
 System.*out*.println("the trump suit is : "+ trumpSuit)**;** //prints trumps  
 }  
 else if(cardArr.get(2).getDigit()==14 || cardArr.get(3).getDigit()==14){ //checks if third and fourth card has an ace  
 trumpSuit = cardArr.get(0).getSuit()**;** //if ace present gives to first and second card suit  
 System.*out*.println("the trump suit is : "+ trumpSuit)**;** //prints trumps  
 }  
 else{  
 trumpSuit = cardArr.get(RandomIndex).getSuit()**;** //randomly selects a trump suit usind random index value calculated  
 System.*out*.println("the trump suit is : "+ trumpSuit)**;** //prints trumps  
 }  
  
 }  
 //if two cards =same suit and other two cards=same suits.  
  
 else if((cardArr.get(0).getSuit().equals(cardArr.get(3).getSuit())) && (cardArr.get(1).getSuit().equals(cardArr.get(2).getSuit()))) {  
 if (cardArr.get(0).getDigit() == 14 || cardArr.get(3).getDigit() == 14) { //checks if first and fourth card have an ace  
 trumpSuit = cardArr.get(1).getSuit()**;** //if present assigns trumps to second or third card suit  
 System.*out*.println("the trump suit is : "+ trumpSuit)**;** //prints trumps  
 }  
 else if (cardArr.get(1).getDigit() == 14 || cardArr.get(3).getDigit() == 14) { //checks iff 2 and 4 card have an ace  
 trumpSuit = cardArr.get(0).getSuit()**;** //if ace is present then assigns trumps to first or third card suit  
 System.*out*.println("the trump suit is : "+ trumpSuit)**;** //prints trumps  
 }  
 else {  
 trumpSuit = cardArr.get(RandomIndex).getSuit()**;** //assigns trumps to random card suit  
 System.*out*.println("the trump suit is : "+ trumpSuit)**;** //prints trumps  
 }  
  
 }  
  
  
 else if((cardArr.get(0).getSuit().equals(cardArr.get(2).getSuit())) && (cardArr.get(1).getSuit().equals(cardArr.get(3).getSuit()))) { //first=third and second=fourth card suit  
 if (cardArr.get(0).getDigit() == 14 || cardArr.get(2).getDigit() == 14) { //checks if ace is present on first or third  
 trumpSuit = cardArr.get(1).getSuit()**;** //if present assigns trumps to second and fourth card suit  
 System.*out*.println("the trump suit is : "+ trumpSuit)**;** ///prints trumps  
 }  
 else if (cardArr.get(1).getDigit() == 14 || cardArr.get(2).getDigit() == 14) { //checks if ace is present on second or fourth  
 trumpSuit = cardArr.get(0).getSuit()**;** //if present assigns trumps to first and third card suit  
 System.*out*.println("the trump suit is : "+ trumpSuit)**;** //prints trumps  
 }  
 else {  
 trumpSuit = cardArr.get(RandomIndex).getSuit()**;** //assigns random index card suit as trumps  
 System.*out*.println("the trump suit is : "+ trumpSuit)**;** //prints trump  
 }  
  
 }  
  
 //checks if two only one equal suit  
 else if(cardArr.get(0).getSuit().equals(cardArr.get(1).getSuit())){ //if first and second same  
 trumpSuit = cardArr.get(1).getSuit()**;** //assigns trumps to either first or second  
 System.*out*.println("the trump suit is : "+ trumpSuit)**;** //prints trumps  
 }  
 else if(cardArr.get(2).getSuit().equals(cardArr.get(3).getSuit())){ //if third and fourth suit same  
 trumpSuit = cardArr.get(3).getSuit()**;** //assigns trumps to 4th card suit  
 System.*out*.println("the trump suit is : "+ trumpSuit)**;** //prints trumps  
 }  
 else if(cardArr.get(0).getSuit().equals(cardArr.get(2).getSuit())){ //if first and third card suit same  
 trumpSuit = cardArr.get(2).getSuit()**;** //assigns trumps to third card suit  
 System.*out*.println("the trump suit is : "+ trumpSuit)**;** //prints trumps  
 }  
 else if(cardArr.get(1).getSuit().equals(cardArr.get(3).getSuit())){ //if second and fourth suit same  
 trumpSuit = cardArr.get(1).getSuit()**;** //assigns trumps to second  
 System.*out*.println("the trump suit is : "+ trumpSuit)**;** //prints trumps  
 }  
 else if(cardArr.get(0).getSuit().equals(cardArr.get(3).getSuit())){ //assigns first and fourth suit same  
 trumpSuit = cardArr.get(3).getSuit()**;** //assigns trump suit to fourth  
 System.*out*.println("the trump suit is : "+ trumpSuit)**;** //prints trumps  
 }  
 else if(cardArr.get(1).getSuit().equals(cardArr.get(2).getSuit())){ //if second and third suit same  
 trumpSuit = cardArr.get(2).getSuit()**;** //assigns trumps to the third suit  
 System.*out*.println("the trump suit is : "+ trumpSuit)**;** //prints trumps  
 }  
  
 //checks if all four cards have a different suit  
 else{  
 trumpSuit = cardArr.get(RandomIndex).getSuit()**;** //gets a random card and assigns its trump suit  
 System.*out*.println("the trump suit is : "+ trumpSuit)**;** //prints trumps  
 }  
  
  
  
  
  
  
 }  
  
  
}