CREATE PT CODE

```
import java.util.Scanner;

public class CreatePT {

	public static Scanner s = new Scanner(System.in); // scanner public static
	boolean on = true;
	public static boolean bj = false;

	public static void main(String[] args) {

		System.out.println("Welcome to Blackjack! \n");

		Deck d = new Deck(); d.fillDeck();
		d.shuffle();
		d.getDeck();
		loadHand(d);

	} // end main

	public static void loadHand(Deck d) {

		Hand dlh = new Hand(d.getDeck().remove(0), d.getDeck().remove(0), true);
		System.out.println("Dealer's Hand: "); dlh.printHand(bj);

		Hand plh = new Hand(d.getDeck().remove(0), d.getDeck().remove(0), false);
		System.out.println("Your Hand: "); plh.printHand(bj);

		if (plh.getSum() == 21 && (dlh.getSum() != plh.getSum())) { System.out.println("BLACKJACK!
			\nYou Win! ");
		} else if (dlh.getSum() == 21 && (dlh.getSum() != plh.getSum())) {
			System.out.println("\nDealer's Hand: "); bj =
			true; dlh.printHand(bj);
			System.out.println("The dealer has blackjack. \nYou Lose! ");
		} else {
			hit(plh, dlh, d);
		}

	} // end loadHand

	public static void hit(Hand plh, Hand dlh, Deck d) {
		boolean hit = true;
		do {
			System.out.println("Press 1 to hit. Press 2 to stay. "); int x =
			s.nextInt(); if (x == 1) {
```

```
                                plh.hit(d, plh);
                                System.out.println("\nYour Hand: "); plh.printHand(bj);
                        } else if (x == 2) { hit =
                                false;
                        } else {
                                System.out.println("Error. Please enter a valid input.");
                        }
                } while (hit && lose(plh, dlh));
 if (lose(plh, dlh)) {
                        dlh.hit(d, plh);
                        System.out.println("\nDealer's Hand: "); dlh.printHand(bj);
                        System.out.println("Your Hand: "); plh.printHand(bj);

                        win(plh, dlh, d);
                }
        } // end hit

        public static void win(Hand plh, Hand dlh, Deck d) { if (lose(plh, dlh) == true) { if (((plh.getSum() >
                dlh.getSum()) && ((plh.getSum()) <= 21))|| (dlh.getSum() >
 21)) {
                                  System.out.println("You Win! ");
                        } else if (plh.getSum() < dlh.getSum() && (dlh.getSum()) <= 21) {
                                System.out.println("You Lose! ");
                        } else if (plh.getSum() == dlh.getSum()) { System.out.println("PUSH ");
                        }
                } // end if
                //System.out.println("\nPress Ctrl + F11 to play again! ");
        } // end win

        public static boolean lose(Hand plh, Hand dlh) { if (on ==
                true) {
                        for (int i = 0; i < plh.getHand().size() - 1; i++) { if
                                (plh.getHand().get(i).getName().equals("Ace")) {
                                plh.ace(plh.getHand().get(i));
                                }
                        }
                        for (int i = 0; i < dlh.getHand().size() - 1; i++) {
                                if (dlh.getHand().get(i).getName().equals("Ace")) {
                                        dlh.ace(dlh.getHand().get(i));
                                }
                        }
                        if (plh.getSum() > 21) {
                                  System.out.println("You Lose! ");
                        //      System.out.println("\nPress Ctrl + F11 to play again! "); on =
                                false;
                        }
                        return on;
                }
```

```
                return false;
        } // end win

} // end class
```

```
import java.util.ArrayList;
import java.util.Random;

public class Deck {

        private ArrayList<Card> deck;

        public Deck() {
 deck = new ArrayList<Card>();

        }

        public void fillDeck() {
 for (int j = 0; j < 4; j++) {
 for (int i = 1; i < 14; i++) {

                                        switch (i) { case 1: deck.add(new Card(11,
                                        "Ace")); break;
                                        case 2: deck.add(new Card(i, "Two"));
                                                break;
                                        case 3: deck.add(new Card(i, "Three"));
                                                break;
                                        case 4: deck.add(new Card(i, "Four"));
                                                break;
                                        case 5: deck.add(new Card(i, "Five"));
                                                break;
                                        case 6: deck.add(new Card(i, "Six"));
                                                break;
                                        case 7: deck.add(new Card(i, "Seven"));
                                                break;
                                        case 8: deck.add(new Card(i, "Eight"));
                                                break;
                                        case 9: deck.add(new Card(i, "Nine"));
                                                break;
                                        case 10: deck.add(new Card(i, "Ten"));
                                                break;
                                        case 11: deck.add(new Card(10, "Jack"));
                                                break;
                                        case 12: deck.add(new Card(10, "Queen"));
                                                break;
```

```
                                        case 13: deck.add(new Card(10, "King"));
                                                 break;

                                        } // end case
                                } // end loop
                        } // end loop
                } // end fillDeck


                public void shuffle() {

                        ArrayList<Card> TMP = new ArrayList<Card>();
                        Random r = new Random();

                        for (int i = 0; i < 52; i++) {
                                        TMP.add(deck.remove(r.nextInt(deck.size())));
                        } // end loop
                        deck = TMP;

                } // end shuffle

                public ArrayList<Card> getDeck () { return deck;
                }

} // end class




import java.util.ArrayList;

public class Hand {

                private ArrayList<Card> hand = new ArrayList<Card>(); private boolean
                playerType;

                public Hand (Card Card1, Carda Card2, boolean playerType) {
                        hand.add(Card1); hand.add(Card2);
                        this.playerType=playerType;
                }

                public int getSum (){ int
                        sum=0;
                        for(int i=0; i<hand.size(); i++){
                         sum         += hand.get(i).getNum();
                        }
                        return sum;
                }
```

```
	public ArrayList<Card> getHand (){ return
		hand;
	}

	public void hit (Deck d, Hand plh){ int
		i=1; if(playerType)
		{	do	{	if	(getSum()<17	&&	playerType){
			hand.add(d.getDeck().get(0));
			d.getDeck().remove(0);
					System.out.println("The dealer was dealt: [" +
hand.get(i).getName() + "]");
				i+=1;
			}
			} while (getSum()<17);
		}
		else { hand.add(d.getDeck().remove(0));
			System.out.println("You were dealt: [" +
plh.getHand().get(hand.size()-1).getName() + "]");
		}
		playerType=false;
	}

	public void ace(Card ace) { if
		(getSum()>21){
		ace.swapAce();
		}
	} // end ace

	public void printHand (boolean bj){

		for (int i=0; i<hand.size(); i++){ if (playerType &&
			i==0 && bj==false){ System.out.print("[?]
			");
			}
			else{
				System.out.print("[" + hand.get(i).getName()+ "] ");
			}
		}
		System.out.println("\n");
	}
} // end class
```

```
public class Card {

	private int num;
	private String name;

	    public Card(int num, String name){

			this.num=num;
			this.name=name;

	} // end card

	public void swapAce() { this.num=1;
	}


public int getNum() {
	return num;
}

public String getName() {
	return name;
}

} // end Card
```