Program 1
This assignment will train you for our class trip to Foxwoods.
Here is a somewhat simplified version of Blackjack as in played in a casino…kinda

Each card has a score based on its rank. However, Jacks, Queens ,and Kings have a score of 10.
An Ace can be either 1 or 11.

So for example the hand
- K of Hearts (10)
- 3 of Clubs (3)
- 7 of Spades (7)

Has a score of 20

The hand
- Ace of Diamonds(1 or 11)
- 7 of Hearts (7)

Can be scored as either 8 or 18

You are playing against the dealer or for this program, the computer.
The object is to draw a hand as close to 21 as possible without going over 21 and to beat the dealer's score.

So for example, if your hand is
- Ace of Diamonds
- 7 of Hearts

Score (18 or 7) – use the better one

And the dealer has
- 3 of Spades
- Jack of Clubs
- 4 of Hearts

Score (17)
You win because you can get 18 but the dealer just 17

So here is how it is played in the casino, with a few simplifications and variations
The “dealer” is the computer. So, you are playing against the computer

1. The dealer will deal you two cards

2. The dealer then deals herself two cards but displays only one for you to see

3. Now it your turn to ask for more cards **one at a time**, but you don’t want to go over 21. You can stop at any time
   Asking for another card is called a “hit.” When you want no more cards, you “stand.”

4. If you go over 21 , you automatically lose and the game is over and you lose

5. If you have exactly 21 you automatically win and the game is over

6. Otherwise, the dealer takes more cards, one at a time, but stops at a score of 17 or higher

7. Now compare scores
   if dealer score is greater than player score dealer wins
   If dealer score is less than player score player wins
   if scores are equal there is a tie

You obviously want the highest score you can get without going over 21.
Scoring a hand is not hard but you have to consider aces.

To score a hand that contains an ace
1. Have a boolean called ace that is set to true if you are dealt an ace

2. Calculate the score using a value of 1 for aces

3. If ace == true and the (score+10) is <= 21 add 10 to the score
   (In a hand with more than one ace, only one can be 11 or you will go over 21)

Here a is output from three games:

```
Your first two cards are
        5 of Hearts
        King of Hearts

Dealer shows one card
        3 of Hearts

 Your turn: Type 0 for a hit any
other number to stand:  0

Your card is
        King of Spades

Your hand is now
        5 of Hearts
        King of Hearts
        King of Spades

 Your score is 25 You lose
```

```
Your first two cards are
        6 of Hearts
        6 of Spades

Dealer shows one card
        2 of Clubs

Your turn: Type 0 for a hit any other
number to stand:  0

Your card is
        2 of Diamonds

Your hand is now
        6 of Hearts
        6 of Spades
        2 of Diamonds

 Another? Type 0 for a hit:  0
Your card is
        4 of Spades

Your hand is now
        6 of Hearts
        6 of Spades
        2 of Diamonds
        4 of Spades

 Another? Type 0 for a hit:  9

Dealer plays and has these cards
        10 of Hearts
        2 of Clubs
        8 of Hearts
------------------
Dealer score 20
Player score 18
------------------
Dealer wins
```

```
Your first two cards are
        8 of Hearts
        4 of Hearts

Dealer shows one card
        8 of Diamonds

 Your turn Type 0 for a hit any
othe number to stand:  0

Your card is
        9 of Hearts

Your hand is now
        8 of Hearts
        4 of Hearts
        9 of Hearts

 Another? Type 0 for a hit:  9

 Your score is 21 You win
>
```

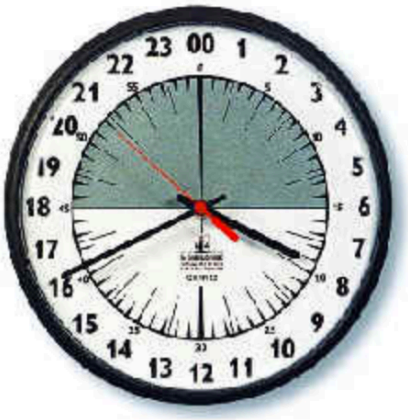
OK…I will help you with this one .
Here is a possible structure for the program. You really just have to fill in the details

```
public class Blackjack
{
        private Card[] dealerCards;  // an array of Card objects
        private Card[] playerCards;  // an array of Card objects

        Deck deck;  //We did this in class

        private int dealerNumCards;  // the number of cards the dealer has
        private int playerNumCards;  //the number of cards the player has

  public Blackjack()  // constructor
  {
    dealerCards  = new Card[15];  // 15 cards is much more than you will need
    playerCards  = new Card[15];
    dealerNumCards= playerNumCards = 0;
    deck = new Deck();
  }

  public void play()
  {
        //Deal and display 2 cards for the player; don't forget to increment
playerNumCards

        // Deal 2 cards to the dealer but display only one of them
          // don't forget to increment dealer NumCards

         // continually ask the player to "hit" or "stand" until the player stands
         // I used 0 for a hit any other number to stand
         // a hit means get another card
        // don't forget to increment playerNumCards  for the player
        // be sure to display the player's cards

         // score the player's hand
        // if the score us over 21 → player loses (and return). →game over
        //If the score == 21 →player wins a(and return) → game over

        // otherwise the dealer takes cards, one at a time, stopping if score is >= 17
        // don't forget to increment dealerNumCards
        // display the dealer's hand and score
        // compare dealer and player scores to determine who wins.
  }
```

```
  public int score (Card[] c, int num) // returns the score of a hand containing num cards
  {
    // c is a hand
   // num is the number of cards in the hand
    }

  public static void main(String[] args)
  {
    Blackjack bj = new Blackjack();
    bj.play();
  }
}
```

That is everything.
Notice that main(..) consists of two lines. Do not put the logic of the program into main(..)

You need the Card class and the Deck class that we made in class.



Program 2

Design a class that models a 24 - hour Clock class. The class should include variables:

hours, minutes.

This is a 24-hour clock so that 13:15 is 1:15:PM.

There should be three constructors:

- a default constructor that sets the time to midnight hour = 0, minute = 0
- a one-argument constructor with parameter for the hour This constructor always sets the minute to 0.
- a two-argument constructor with parameters for the hour and minute

Methods should be

- int getHour();
  //returns the hour
- int getMinute();
  //returns the minute
- String toString()
  //returns a String in the form x:xx PM eg. "2:53 PM" 0r "12:15 AM"
- void incrementTimer();
  //basic incrementer adds 1 minute to the time
- void incrementTimer(int x);
  //overloaded method ; adds x minutes to the time
- void SetTime(int h, int m);
  // Sets the time.
  // All values must all be range-checked to make sure they are valid

// For example, 3:78 is invalid.
// If invalid data is entered a message should be issued and end the program

Notes:

1. You'll have to pay careful attention to the wraparound feature. For instance, if you add 1000 minutes to 15:24 the new time is 8:04.

2. Write a second class (DemoTimer) that uses your Timer class

- The main method of DemoTimer should
- instantiate a timer (time) with initial time 23:54
- print the current time (as 11:54 PM)
- add one minute to the time
- print the new time (as 11:55 PM)
- add 190 minutes to the current time
- print the new time
- prompt for a new time (hour and minute)
- set time to the new time
- print the new time

Output: (Program prompts are in red, User input is blue)

Current time is 11:54 PM (Initial time)
Current time is 11:55 PM (After one minute)
Current time is 3:05 AM (After 190 minutes)
Enter hour: 15
Enter minute: 23
Current time is 3:23 PM
Note: Data is entered using the 24 hour model.

You need to save each class in a separate file: Timer.java and DemoTimer.java