COW - Randoms

Level 1

Fill in the code for the RandomNumbers Class. Each method in the RandomNumbers Class should print out a series of 30 randomly picked integers with the given range (inclusive):

```
random1() - a range of 0 to 10
random2() - a range of 0 to 100
random3() - a range of 10 to 110
random4() - a range of 12 to 578
random5() - a range of -142 to -13
random6() - a range of -300 to 412
```

Level 2

Fill in the following methods in the RandomStuff Class. Each method in the RandomStuff Class should return a random item from the indicated set:

Hint – set up a static array of the given items, pick a random index, and use the random index to pick a random element in the array

`randomLowerCaseLetter():`	`lowercase letter`
`randomUpperCaseLetter():`	`uppercase letter`
`randomSpecialCharacter():`	`special character`
`randomNumberCharacter():`	`one digit number (char)`
`randomLetter():`	`lowercase, uppercase, special, number`
`randomName():`	`names of AP Computer Science students`

Level 3

Fill in the following methods in the RandomStuff Class.

`randomString()`
returns a random String of length 3

`randomString(int n)`
returns a random String of length n

`randomStringNumber(int n)`
returns a random String composed only of number characters of length n. The first numerical character should not be a ‘0’ and there should be a 50% chance that a ‘-‘ is placed in front

Level 4

Slot Machine

In this you need to program one method in the Slot class:

Name: spin
Input: Nothing
Output: Nothing
Action: sets randomNumber to a random value between 1 and 9

In the SlotMachine class you need to program the following methods:

Name: spinMachine
Input: Nothing
Output: amountWon
Action: this should first spin each slot. It should then use the getNumber method of each slot to determine how much money was won. If none of the slots are equal then it returns the largest number. If two and only two are equal then it should return that number multiplied by 5. If three are equal then it should return 10 times whatever number you have three of. There are two exceptions to these rules which are the numbers 1 and 9. Any time a 1 shows up then the player wins nothing. If a single 9 shows up then the player wins 100. Two nines results in a the player winning 1000. Three nines results in the player winning 10000.

Level 5

In the Wheel class you need to program the following methods:

Name: spin
Input: Nothing
Output: Nothing
Action: sets result to a value between 0 and 36

Name: isEven
Input: Nothing
Output: boolean
Action: returns whether or not result is even. If result it a zero then it return false.

Name: isOdd
Input: Nothing
Output: boolean
Action: returns whether or not result is odd. If result it a zero then it return false.

Name: whichThird
Input: Nothing
Output: int
Action: Returns and int depending on what range of numbers result is in. If result is between 1 and 12 inclusive then it returns a 1. If result is between 13 and 24 inclusive then it returns a 2. If result is between 25 and 36 inclusive then it returns a 3. If result is 0 then it returns a 0.

In the Roulette class you need to program the following methods:

Name: addRandomNumber
Input: Nothing
Output: Nothing
Action: Adds a random number from 1 to 36 to the numberBets ArrayList

Name: spinWheel()
Input: nothing
Output: nothing
Action: in this you need to first spin the wheel object theWheel then store the result in the result variable. Then determine how much was won using theWheel methods and what is stored in amountOnOdd, amountOnEven, amountOn1stThird, amountOn2ndThird, amountOn3rdThird and numberBets. Even and Odd results in getting 2 times that amount added to winnings. Thirds results in getting 3 times that amount added to winnings. And each number in numberBets that matches the result adds 36 to winnings. The winnings should be stored in the winnings variable.

Level 6

In the Dealer class you need to program the following methods:

Name: dealCard
Input: Nothing
Output: Card
Action: return a randomly generated card whose value is between 2 and 14 inclusive and randomly either black or red.

In the Card class you need to program the following methods:

Name: getValue
Input: Nothing
Output: int
Action: return the value of the card. Anything 10 or less is equivalent to its own value. An 11, 12, and 13, result in a 10 being returned. A 14 results in an 11.

In the Hand class you need to program the following methods:

Name: playHand
Input: Nothing
Output: Nothing
Action: calls hit while its score is less than 16

Name: updateScore
Input: nothing
Output: nothing
Action: updates the value stored in score by analyzing all the cards stored in the ArrayList cards