**Part 1:**

**public** **class** mattest {

**private** **int**[][] mat;

**private** **int** len;

**private** **int** width;

mattest() {

mat = **new** **int**[][]{{3,7,2,1},{4,3,4,4},{5,1,3,2}};

setDimensions();

}

mattest(**int**[][] x) {

mat = x;

setDimensions();

}

**private** **void** setDimensions() {

len = mat.length;

**if** (len > 0) {

width = mat[0].length;

}

}

**public** **void** setVal(**int** x, **int** y, **int** z) {

**if** ((x < len) && (y < width)) {

mat[x][y] = z;

}

}

**public** **int** getVal(**int** x, **int** y) {

**return** mat[x][y];

}

**public** **int**[] f() {

**int**[] x = **new** **int**[width];

**for** (**int** k = 0; k < width; k++) {

**int** s = 0;

**for** (**int** j = 0; j < len; j++) {

s += mat[j][k];

}

x[k] = s;

}

**return** x;

}

**public** **void** g() {

**if** (len == width) {

**for** (**int** k=0;k<=len/2;k++) {

**for** (**int** j=0;j<=width/2;j++) {

**int** t = mat[k][j];

mat[k][j]=mat[len-j-1][width-k-1];

mat[len-j-1][width-k-1]=t;

}

}

}

}

**public** String toString() {

String s = "";

**for** (**int** x = 0; x < len; x++) {

**for** (**int** y = 0; y < width; y++) {

s += mat[x][y]+" ";

}

s += '\n'; //this is a new line

}

**return** (s);

}

}

**public** **static** **void** main(String[] args) {

mattest m = **new** mattest();

//Q1: What is printed below??

System.***out***.println(m);

//Q2: What is printedbelow?

System.out.println(m[2,0]);

m.setVal(1, 3, 8);

//Q3: What is printed below??

System.***out***.println(m);

//Q4: What is printed below?

System.***out***.println(m.getVal(1,2));

**int**[][] k = {{3,2,4},{7,1,3},{8,4,2}};

mattest m2 = **new** mattest(k);

//Q5: What is printed below?

System.***out***.println(m2);

//Q6: What is printed below?

System.***out***.println(Arrays.*toString*(m.f()));

m2.g();

//Q7: What is printed below?

System.***out***.println(m2);

}

}

**public** **class** Student {

**private** String first;

**private** String last;

**private** **int** score;

**public** Student(String f, String l,**int** s) {

first = f;

last=l;

score=s;

}

**public** **int** getScore() {

**return** score;

}

**public** String getName() {

String s = first + " "+last;

**return**(s);

}

**public** String toString() {

String str = "";

str += first + " "+ last + ": "+score;

**return**(str);

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** **class** Classes {

**private** **int** coursenum;

**private** **double** ave;

**private** Student[] roster;

Classes(**int** c, Student[] s){

coursenum = c;

roster = s;

findAve();

}

**private** **void** findAve() {

**int** sum = 0;

**for** (**int** x = 0; x < roster.length; x++) {

sum += roster[x].getScore();

}

ave = sum/roster.length;

}

**public** String f() {

**int** b = 0;

**int** s = roster[0].getScore();

**for** (**int** x = 1; x < roster.length; x++) {

**if** (roster[x].getScore()> s) {

s = roster[x].getScore();

b= x;

}

}

**return**(roster[b].toString());

}

**public** String toString() {

String s = "CIS"+coursenum +"\n";

**for** (**int** x=0;x<roster.length;x++) {

s+= roster[x] +"\n";

}

s+= "Ave: "+ave;

**return**(s);

}

}

**public** **static** **void** main(String[] args) {

Student[] x = { **new** Student("Anne","Brown",90),

**new** Student("Ben","Jones",75),

**new** Student("Sam","Smith",95),

**new** Student("Taylor","Guda",80),

**new** Student("Jake","Kelly",85)};

Classes c = **new** Classes(181,x);

//Q8: What is printed below?

System.***out***.println(c);

//Q9: What is printed below?

System.***out***.println(c.f());

**Crazy amount of code, but questions aren’t terrible!!!**

**public** **class** Card {

**private** **int** num;

**private** **char** suit;

Card(**int** n, **char** s) {

num = n;

suit = s;

}

**public** **int** getNum() {

**return**(num);

}

**public** **char** getSuit() {

**return**(suit);

}

**public** String toString() {

String s = num +""+ suit+" ";

**return**(s);

}

}

**public** **class** Deck {

**private** Card[] deck;

Deck() {

deck = **new** Card[52];

**char**[] suits = {'C','S','D','H'};

**int** index = 0;

**for** (**char** s: suits) {

**for** (**int** i = 1; i <= 13; i++) {

deck[index] = **new** Card(i,s);

index++;

}

}

}

**public** Card getCard() {

Random r= **new** Random();

**int** x = r.nextInt(deck.length);

Q10: Fill in the blank so that random card x is returned from the deck

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}

**public** String toString() {

String str="";

**for** (Card i: deck) {

str += i.toString() + " ";

}

**return**(str);

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** **class** Player {

**private** String name;

**public** Card[] hand;

Player(String s) {

name = s;

}

**public** **int** getHighest() {

**int** s = -1;

**for** (**int** x = 0; x < hand.length; x++) {

**if** (hand[x].getNum() > s) {

s = hand[x].getNum();

}

}

**return**(s);

}

**public** String toString() {

String s = "";

**for** (Card x:hand) {

s += x.toString();

}

s += "\n";

**return**(s);

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** **class** CardGame {

**private** Deck deckofcards;

**public** Player[] players;

**private** **int** numofcards;

**public** CardGame(String[] names, **int** n) {

deckofcards = **new** Deck();

numofcards = n;

players = **new** Player[names.length];

**for** (**int** x = 0; x < names.length; x++) {

players[x] = **new** Player(names[x]);

players[x].hand = **new** Card[numofcards];

}

}

**public** **void** DealHands() {

**for** (**int** x = 0; x <players.length; x++ ) {

**for** (**int** y = 0; y < numofcards; y++) {

//Q11: Fill in the line below so that the player gets a card using the Deck's method

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}

}

}

**public** **int** findBest() {

**int** best = -1; //holds the highest score of the best player

**int** i = -1; //holds the index of the best player

**for** (**int** x = 0; x < players.length; x++) {

//Q12: Fill in the lines of code below so that you are finding the player with the highest card (using the Player’s getHighest method)

**if** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ {

i = x;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}

}

**return**(i);

}

**public** String toString() {

String s = "";

**for** (Player x: players) {

s += x.toString() + " ";

}

**return** s;

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** **static** **void** main(String[] args) {

//Q13: Create a CardGame Object, with at least 3 players, and name the CardGame variable g

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

g.DealHands();

System.***out***.println(g);

**int** b = g.findBest();

System.***out***.println(g.players[b]);

}