Labs

(1) Level A	(2) Level B	(2) Level C
Drawing Shapes	Card Game Objects	Class Grades
	Line Lab	Connect Four

Drawing Shapes

Directions

Create a program that allows the user to draw lines, filled rectangles and filled ovals to the screen. For this lab you will be working from a skeleton project where I left comments letting you know what needs to be done.

First copy the drawing shapes folder to your drive. Open the project and complete the files by following the instructions listed in each file. Complete the files in this order: DrawingLine, DrawingRectangle, DrawingOval, DrawingWindow, MainFile.

Rubric (counts as 1 other grade)

Points	Task
15	Drawing Line
15	Drawing Rectangle
15	Drawing Oval
25	Drawing Window
20	Main
10	Formatting and neatness

Card Game Objects

Directions

Create a program that allows the user to draw cards from a deck, view their hand, discard, and to shuffle discarded cards back into the deck. . For this lab you will be working from a skeleton project where I left comments letting you know what needs to be done.

First copy the Card Objects folder to your drive. Open the project and complete the files by following the instructions listed in each file. Complete the files in this order: PlayingCard, Deck, Hand, MainFile.

Rubric (counts as 1 minor grade)

Points	Task
20	Playing Card Class
20	Deck Class
20	Hand Class
20	Main file
20	Formatting and neatness

Line Lab

Directions

Create a program that allows the user to manage two lines and access data about them. The program will start by asking the user to enter the two points for each line. Next the program will create two Line objects from the four points.

Once the lines have been entered the program will give the user a menu with the following six options:

- View Line 1
  - Calls the toString() for line 1
- View Line 2
  - Calls the toString() for line 2
- Change line 1 – Prints the data for line 1 and then asks the user what they want to change.
  - Change Point 1 – changes the 1st point
  - Change Point 2 – changes the 2nd point
  - Change Both – changes both points
  - Cancel
- Change line 2 – Prints the data for line 2 and then asks the user what they want to change.
  - Change Point 1 – changes the 1st point
  - Change Point 2 – changes the 2nd point
  - Change Both – changes both points
  - Cancel
- View Intersection
  - Tells the user if the lines intersect or not. When the lines intersect the user where be told where they intersect.

Equations

slope = (y2-y1) / (x2-x1)

y = mx + b

Coding:

MyPoint

	Description
Constructors	
MyPoint(int x, int y)	Sets the values of x and y
Private Attributes	
int x	The x value of the point
int y	The y value of the point
Methods	
int getX()	Returns the value of x
int getY()	Returns the value of y
void setX(int x)	Changes the value of x
void setY(int y)	Changes the value of y
String toString()	Returns a String containing the data of the point. Example: "(5,3)"

Line

	Description
Constructors	
Line(MyPoint p1, MyPoint p2)	Receives two points and sets its data based on the values of the two points.
Private Attributes	
MyPoint p1	Stores the 1st point of the line
MyPoint p2	Stores the 2nd point of the line
double slope	Stores the slope of the line
double xInt	Stores the x intercept for the line
double yInt	Stores the y intercept for the line
Methods	
int getX()	Returns the value of x
int getY()	Returns the value of y
void setX(int x)	Changes the value of x
void setY(int y)	Changes the value of y
String toString()	Returns a String containing the data for the line. (Both points, slope, intercepts and line equation)

Main File

Methods	Description
void main(String[] args)	Uses MyPoint and Line Objects to complete the tasks of the program.
MyPoint makePoint()	Gathers data from the user and creates/returns a MyPoint made from the entered data.

Rubric (counts as 1 minor grade)

Points	Task
10	makePoint method
30	MyPoint class
30	Line Class
20	Function of the program
10	Formatting and neatness

Class Grades

Directions

Create a program allows the user to enter grades for one of his or her classes. The program will start by getting the name of the class and what type it is (academic Pre-AP or AP).

After the information for the class has been gathered, the program will create a StudentClass with using the data. The program will then come up to a menu with the following options: add assignment, remove assignment, modify assignment, view class information and exit. Each option retrieves and/or modifies the data in the StudentClass object.

Menu Options:

- Add Assignment
  - Gets the name, grade and category of the assignment
  - Adds the assignments to the StudentClass
- Remove Assignment
  - Prints a menu with all the assignments, their scores and categories
  - Allows the user to remove one of the assignments or cancel
- Modify Assignment
  - Prints a menu with all the assignments, their scores and categories
  - Allows the user to modify one of the assignments or cancel
  - When an assignment is selected it will ask what they want to change
    - Name
    - Grade
    - Category
    - Cancel
- CancelView Class Information
  - Displays the name and type of the class
  - Displays the student's minor, other and major averages
  - Displays the student' grade in the class
- Exit – Closes the program

Grade Break downs:

	Academic	Pre-AP	AP
Other	15	10	10
Minor	30	30	20
Major	55	60	70

Coding:

Assignment

	Description
Constructors	
Assignment(String assignmentName, double grade, int category)	Sets assignmentName, grade and category
Public Static Final Attributes	
int OTHER	Stores 0 and is used for category
int MINOR	Stores 1 and is used for category
int MAJOR	Stores 2 and is used for category
Private Attributes	
String assignmntName	Stores the name of the assignment
double grade	Stores the grade of the assignment
int category	Stores the grading category of the assignment
Methods	
int getCategory()	Returns the value for category
double getGrade()	Returns the value for grade
String getAssignmentName()	Returns the value for assignmentName
void setCategory(int category)	Changes the value of category
void setGrade(double grade)	Changes the value of grade
void setAssignmentName(String assignmentName)	Changes the assignmentName

StudentClass

	Description
Constructors	
StudentClass(String className, int type)	Sets className, type, majorWeight, minorWeight, otherWeight and creates the assignments ArrayList.
Public Static Final Attributes	
int ACADEMIC	Stores 0 and is used for type
int PRE_AP	Stores 1 and is used for type
int AP	Stores 2 and is used for type
Private Attributes	
double majorWeight	Stores the percentage that major grades are worth
double minorWeight	Stores the percentage that minor grades are worth
double otherWeight	Stores the percentage that other grades are worth
ArrayList<Assignment> assignments	Stores all the assignments.
String className	Stores the name of the class.
int type	Stores the type of the class
Methods	
String getClassName()	Returns the value of className
ArrayList<Assignment> getAssignments()	Returns the assignments ArrayList
double getMinorWeight()	Returns the value of minorWeight
double getMajorWeight()	Returns the value of majorWeight
double getOtherWeight()	Returns the value of otherWeight
int getType()	Returns the value of type
void addAssignment(Assignment a)	Adds the given assignment to the assignments ArrayList
double average()	Returns the average for the class
double minorAverage()	Calculates and returns the minor average.
double majorAverage()	Calculates and returns the major average.
double otheAverage()	Calculates and returns the other average.

Main File

Methods	Description
void main(Strng[] args)	Uses the StudentClass and Assignment Objects to complete the tasks of the program.

Rubric (counts as 2 minor grades)

Points	Task
30	Assignment Class
30	StudentClass Class
30	Main
10	Formatting and neatness

Connect Four

Directions

Create a text based game of connect four. For this lab you will need to create the following two files: MainFile and ConnectFourGame.

The MainFile file will create a single ConnectFourGame object and use its methods to allow two users to play a single game of connect four. The game will stop when one player wins or the game board is full. If a player makes an invalid move he/she will be prompted for a new move.

Coding:

ConnectFourGame

	Description
Constructors	
ConnectFourGame	Creates the board Sets all the board locations to EMPTY
Public Static Attributes	
int PLAYING	Stores 0 and is used for the status method
int RED_WINS	Stores 1 and is used for status
int BLACK_WINS	Stores 2 and is used for status
int DRAW	Stores 3 and is used for status
int RED	Stores 4 and is used for the board
int BLACK	Stores 5 and is used for the board
int EMPTY	Stores 6 and is used for the board
Attributes	
int board[][]	Stores the data of the game board
Methods	
boolean dropPiece (int column, int player)	Attempts to place player's color in the lowest row of the given column. Returns true if the placement was successful and false otherwise. column should be 0 to 6 player should be RED or BLACK
int status()	Returns the state of the game PLAYING, BLACK_WINS, RED_WINS, or DRAW
boolean columnFull(int column)	Returns true if the column is full and false if it is not.
void draw()	Prints the board to the screen. The output should look similar to the following board: | | | | | | | | | | | | | | | | | | | | | R | | | | | | | | B | | | | | | | | R | | B |

	| R | B | R | B | R | R | B |
int getSpot(int row, int col)	Returns the value at the given row and column, -1 if the location is invalid Note: This method will be used in future labs.

Main:

	Description
Methods	
Main	Uses a ConnectFourGame object to play Connect Four

Extra Credit Part 1

Allow the user to player versus another player or a random moving computer. The computer will always make a valid move.

Extra Credit Part 2

The computer will always block when the player has 3 in a row.

Rubric (counts as 2 minor grades)

Points	Task
10	Connect Four Game (Constructor and Attributes)
10	dropPiece
15	status
10	getSpot
10	columnFull
15	draw
20	Main File
10	Formatting and neatness
10	Extra Credit 1
15	Extra Credit 2