**Programming CS 1083 LAB BACK UPS**

**LAB 0**

/\*\*

\* Auto Generated Java Class.

\*/

public class Initials {

public static void main(String[] args) {

System.out.println( "Lab 0 written by Grecia Alieth Gonzalez");

System.out.println(" GGGGGG AA GGGGG ");

System.out.println(" G G A A G G ");

System.out.println(" G G A A G G ");

System.out.println(" G AAAAAAAA G ");

System.out.println(" G GGGG A A G GGGG ");

System.out.println(" GGGGG A A GGGGG ");

}

/\* ADD YOUR CODE HERE \*/

}

**LAB 1**

/\*This program prints the first verse of "Mary Had a Little Lamb"

and the first three verses of "This Old Man".\*/

public class Poems {

/\* This main method prints my name, as well as call the two methods

"MaryLittleLamb()" and "Oldman()"\*/

public static void main(String[] args) {

System.out.println(" Lab 1 written by Grecia Alieth Gonzalez");

System.out.println();

MaryLittleLamb();

System.out.println();

Oldman();

}

//This method prints the first verse and titele of "Mary Had a Little Lamb"

public static void MaryLittleLamb( ){

System.out.println("Title :Mary Had a Little Lamb Song");

System.out.println();

Mary();

Little();

Mary();

Snow();

}

//This method prints the first line and the third line of "Mary Had a Little Lamb"

public static void Mary(){

System.out.println("Mary Had a Little Lamb");

}

//This method prints the second line of "Mary Had a Little Lamb"

public static void Little() {

System.out.println("Little Lamb, Little Lamb");

}

//This method prints the fourth line of "Mary Had a Little Lamb"

public static void Snow() {

System.out.println("Whose fleece was white as snow");

}

//This method prints the title and the first three verses of "This Old Man"

public static void Oldman() {

System.out.println("Title: This Old Man Song");

System.out.println();

Oldmanfirst();

Oldmansecond();

Oldmanthird();

}

//This method prints the first verse of "This Old Man"

public static void Oldmanfirst() {

System.out.println("This Old Man he played one");

System.out.println("He played nick nack on my drum");

NickNack();

}

//This method prints the second verse of "This Old Man"

public static void Oldmansecond(){

System.out.println("This Old Man he played two");

System.out.println("He played nick nack on my shoe");

NickNack();

}

//This method prints the third verse of "This Old Man"

public static void Oldmanthird(){

System.out.println("This Old Man he played three");

System.out.println("He played nick nack on my tree");

NickNack();

}

//This method prints the repeting phrase of "This Old Man"

public static void NickNack( ){

//This method prints out the repeating verse of the song

System.out.println("With a nick nack paddy whack");

System.out.println("Give a dog a bone");

System.out.println("This old man came rolling home");

System.out.println();

}

}

**LAB 2:**

/\* We need a Java utility (the Scanner class) for keyboard input\*/

import java.util.\*;

public class Coins {

public static final Scanner CONSOLE= new Scanner(System.in);

/\* Prints out the title and author as well as calls for 2 methods, "TotalCentsandVal" and "SpecificCoins".\*/

public static void main(String[] args) {

DecimalFormat dollarsAndCents = new DecimalFormat("$#0.00");

return dollarsAndCents.format(amount);

System.out.println("Lab 2 written by Grecia Alieth Gonzalez ");

TotalCentsandVal();

SpecificCoins();

}

/\*This method asks the user to type the value of 4 variables(quarters,nickles,dimes,and pennies)

to convert them into a value in dollars and count the total number of coins .\*/

public static void TotalCentsandVal ( ) {

/\*declare variables\*/

int quarters;

int dimes;

int nickles;

int pennies;

int totalcents;

double dollars;

System.out.println("Enter number of quarters:");

quarters = CONSOLE.nextInt();

System.out.println("Enter number of dimes:");

dimes = CONSOLE.nextInt();

System.out.println("Enter number of nickles:");

nickles = CONSOLE.nextInt();

System.out.println("Enter number of pennies:");

pennies = CONSOLE.nextInt();

totalcents = quarters + dimes + nickles + pennies;

dollars = ((quarters \* 0.25) + (dimes \* 0.10) + (nickles \*0.05) + (pennies \* 0.01));

System.out.println(" Total number of coins is " + totalcents);

System.out.println(" Total value in dollars is $ " + dollars);

}

/\*This method asks the user to type an amount of change in order to determine the number of quarters,

dimes, nickles, and pennies that sum up to that value.\*/

public static void SpecificCoins ( ) {

/\*declare variables\*/

int cents;

int quarters;

int dimes;

int nickles;

int pennies;

int remdrquarters;

int remdrdimes;

int remdrnickles;

System.out.println( "Enter the total amount of cents:");

cents = CONSOLE.nextInt();

quarters = cents / 25;

remdrquarters = cents % 25;

dimes = remdrquarters / 10;

remdrdimes = remdrquarters % 10;

nickles = remdrdimes / 5;

remdrnickles = remdrdimes % 5;

pennies = remdrnickles / 1;

System.out.println(" Total number of quarters " + quarters);

System.out.println(" Total number of dimes " + dimes);

System.out.println(" Total number of nickles " + nickles);

System.out.println(" Total number of penneies " + pennies);

}

}

**LAB 3**

/\* We need a Java utility (the Scanner class) for keyboard input\*/

import java.util.\*;

public class FutureValues {

public static final Scanner CONSOLE= new Scanner(System.in);

/\* Prints out the title and author as well as calls for 2 methods, "ValueAccount" and "ValueAnnuity".

The main console allows user to type in principal amount,interest rate, years ccompleted,

then prints account value. Then allows user to tpe in yearly payment,interest rate and number of years. Then

prints out future value.\*/

public static void main(String[] args) {

System.out.println("Lab 3 written by Grecia Gonzalez");

System.out.println("Principal Amount:");

double prncpl = CONSOLE.nextInt();

System.out.println("Interest Rate:");

double rate = CONSOLE.nextInt();

System.out.println("Years Completed:");

double yr = CONSOLE.nextInt();

System.out.println("Account Value"+ valueAccount(prncpl,rate,yr));

System.out.println("Yearly Payment:");

double yp = CONSOLE.nextInt();

System.out.println("Interest Rate:");

double ir = CONSOLE.nextInt();

System.out.println("Number of Years:");

double ny = CONSOLE.nextInt();

System.out.println("Future Value:" + valueAnnuity(yp,ir,ny));

valueAccount(prncpl,rate,yr);

valueAnnuity(yp,ir,ny);

}

/\*This method calculates future value using compound interes\*/

public static double valueAccount(double prncpl, double rate, double yr){

double x = prncpl\*Math.pow(1+(rate/100),yr);

return x;

}

/\*This method calculates future value based on annuity\*/

public static double valueAnnuity(double yp, double ir,double ny){

double x = yp\*(Math.pow(1+(ir/100),ny)-1)/(ir/100);

return x;

}

}

**LAB 4**

**Draw Panel**

/\*

Stuart Reges and Marty Stepp

February 24, 2007

Changes by Tom Bylander in 2010 (no anti-alias, repaint on sleep)

Changes by Tom Bylander in 2012 (track mouse clicks and movement)

Changes by Tom Bylander in 2013 (fix bug in tracking mouse clicks)

Changes by S. Robbins in 2014 (getters for width and height)

Changes by S. Robbins in 2014 (addKeyListener added)

Changes by S. Robbins in 2014 (catch exception on default close so that it works in an applet)

Changes by S. Robbins in 2015 (buffer key events)

Changes by S. Robbins in 2015 (show mouse status by default is off)

The DrawingPanel class provides a simple interface for drawing persistent

images using a Graphics object. An internal BufferedImage object is used

to keep track of what has been drawn. A client of the class simply

constructs a DrawingPanel of a particular size and then draws on it with

the Graphics object, setting the background color if they so choose.

To ensure that the image is always displayed, a timer calls repaint at

regular intervals.

\*/

import java.awt.\*;

import java.awt.event.\*;

import java.awt.image.\*;

import javax.swing.\*;

import javax.swing.event.\*;

import java.util.ArrayList;

public class DrawingPanel implements ActionListener {

private static final String versionMessage =

"Drawing Panel version 1.1, January 25, 2015";

private static final int DELAY = 100; // delay between repaints in millis

private static final boolean PRETTY = false; // true to anti-alias

private static boolean showStatus = false;

private static final int MAX\_KEY\_BUF\_SIZE = 10;

private int width, height; // dimensions of window frame

private JFrame frame; // overall window frame

private JPanel panel; // overall drawing surface

private BufferedImage image; // remembers drawing commands

private Graphics2D g2; // graphics context for painting

private JLabel statusBar; // status bar showing mouse position

private volatile MouseEvent click; // stores the last mouse click

private volatile boolean pressed; // true if the mouse is pressed

private volatile MouseEvent move; // stores the position of the mouse

private ArrayList<KeyInfo> keys;

// construct a drawing panel of given width and height enclosed in a window

public DrawingPanel(int width, int height) {

this.width = width;

this.height = height;

keys = new ArrayList<KeyInfo>();

image = new BufferedImage(width, height, BufferedImage.TYPE\_INT\_ARGB);

statusBar = new JLabel(" ");

statusBar.setBorder(BorderFactory.createLineBorder(Color.BLACK));

statusBar.setText(versionMessage);

panel = new JPanel(new FlowLayout(FlowLayout.CENTER, 0, 0));

panel.setBackground(Color.WHITE);

panel.setPreferredSize(new Dimension(width, height));

panel.add(new JLabel(new ImageIcon(image)));

click = null;

move = null;

pressed = false;

// listen to mouse movement

MouseInputAdapter listener = new MouseInputAdapter() {

public void mouseMoved(MouseEvent e) {

pressed = false;

move = e;

if (showStatus)

statusBar.setText("moved (" + e.getX() + ", " + e.getY() + ")");

}

public void mousePressed(MouseEvent e) {

pressed = true;

move = e;

if (showStatus)

statusBar.setText("pressed (" + e.getX() + ", " + e.getY() + ")");

}

public void mouseDragged(MouseEvent e) {

pressed = true;

move = e;

if (showStatus)

statusBar.setText("dragged (" + e.getX() + ", " + e.getY() + ")");

}

public void mouseReleased(MouseEvent e) {

click = e;

pressed = false;

if (showStatus)

statusBar.setText("released (" + e.getX() + ", " + e.getY() + ")");

}

public void mouseEntered(MouseEvent e) {

// System.out.println("mouse entered");

panel.requestFocus();

}

};

panel.addMouseListener(listener);

panel.addMouseMotionListener(listener);

new DrawingPanelKeyListener();

g2 = (Graphics2D)image.getGraphics();

g2.setColor(Color.BLACK);

if (PRETTY) {

g2.setRenderingHint(RenderingHints.KEY\_ANTIALIASING, RenderingHints.VALUE\_ANTIALIAS\_ON);

g2.setStroke(new BasicStroke(1.1f));

}

frame = new JFrame("Drawing Panel");

frame.setResizable(false);

try {

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); // so that this works in an applet

} catch (Exception e) {}

frame.getContentPane().add(panel);

frame.getContentPane().add(statusBar, "South");

frame.pack();

frame.setVisible(true);

toFront();

frame.requestFocus();

// repaint timer so that the screen will update

new Timer(DELAY, this).start();

}

public void showMouseStatus(boolean f) {

showStatus = f;

}

public void addKeyListener(KeyListener listener) {

panel.addKeyListener(listener);

panel.requestFocus();

}

// used for an internal timer that keeps repainting

public void actionPerformed(ActionEvent e) {

panel.repaint();

}

// obtain the Graphics object to draw on the panel

public Graphics2D getGraphics() {

return g2;

}

// set the background color of the drawing panel

public void setBackground(Color c) {

panel.setBackground(c);

}

// show or hide the drawing panel on the screen

public void setVisible(boolean visible) {

frame.setVisible(visible);

}

// makes the program pause for the given amount of time,

// allowing for animation

public void sleep(int millis) {

panel.repaint();

try {

Thread.sleep(millis);

} catch (InterruptedException e) {}

}

// close the drawing panel

public void close() {

frame.dispose();

}

// makes drawing panel become the frontmost window on the screen

public void toFront() {

frame.toFront();

}

// return panel width

public int getWidth() {

return width;

}

// return panel height

public int getHeight() {

return height;

}

// return the X position of the mouse or -1

public int getMouseX() {

if (move == null) {

return -1;

} else {

return move.getX();

}

}

// return the Y position of the mouse or -1

public int getMouseY() {

if (move == null) {

return -1;

} else {

return move.getY();

}

}

// return the X position of the last click or -1

public int getClickX() {

if (click == null) {

return -1;

} else {

return click.getX();

}

}

// return the Y position of the last click or -1

public int getClickY() {

if (click == null) {

return -1;

} else {

return click.getY();

}

}

// return true if a mouse button is pressed

public boolean mousePressed() {

return pressed;

}

public synchronized int getKeyCode() {

if (keys.size() == 0)

return 0;

return keys.remove(0).keyCode;

}

public synchronized char getKeyChar() {

if (keys.size() == 0)

return 0;

return keys.remove(0).keyChar;

}

public synchronized int getKeysSize() {

return keys.size();

}

private synchronized void insertKeyData(char c, int code) {

keys.add(new KeyInfo(c,code));

if (keys.size() > MAX\_KEY\_BUF\_SIZE) {

keys.remove(0);

// System.out.println("Dropped key");

}

}

private class KeyInfo {

public int keyCode;

public char keyChar;

public KeyInfo(char keyChar, int keyCode) {

this.keyCode = keyCode;

this.keyChar = keyChar;

}

}

private class DrawingPanelKeyListener implements KeyListener {

int repeatCount = 0;

public DrawingPanelKeyListener() {

panel.addKeyListener(this);

panel.requestFocus();

}

public void keyPressed(KeyEvent event) {

// System.out.println("key pressed");

repeatCount++;

if ((repeatCount == 1) || (getKeysSize() < 2))

insertKeyData(event.getKeyChar(),event.getKeyCode());

}

public void keyTyped(KeyEvent event) {

}

public void keyReleased(KeyEvent event) {

repeatCount = 0;

}

}

}

**Hyperbola**

/\* This program draws two parabolas with grid lengths defined by LOWER\_GRID\_LENGTH and UPPER\_GRID\_LENGTH inside a

\* DrawingPanel with dimensions defined by PANEL\_SIZE.

\*/

import java.awt.\*;

public class Hyperbola {

//Dimension Requirements: hyperbola DrawingPanel should be 512 pixels wide and 512 pixels tall.

//The hyperbola in the lower left-hand corner has grid lines separated by 32 pixels, and the hyperbola in the

//upper right-hand corner has grid lines separated by 8 pixels

//Use of 'private static final' bc these dimensions stay the same

private static final int PANEL\_SIZE = 512;

private static final int LOWER\_GRID\_LENGTH = 32;

private static final int UPPER\_GRID\_LENGTH = 8;

public static void main(String[] args) {

System.out.println("Lab 4 written by Grecia Alieth Gonzalez");

DrawingPanel panel = new DrawingPanel(PANEL\_SIZE, PANEL\_SIZE);

Graphics g = panel.getGraphics();

//for loop of the lower left hyperbola

for(int i = 0; i <= PANEL\_SIZE; i+=LOWER\_GRID\_LENGTH)

g.drawLine(0, i, i, PANEL\_SIZE);

//For loop of the upper right hyperbola

for(int i = PANEL\_SIZE; i >= 0; i-=UPPER\_GRID\_LENGTH)

g.drawLine(PANEL\_SIZE, i, i, 0);

}

}

**StarwarsName**

/\* This program prompts the user for their first and last name as well as his or her mother's maiden name, the city

in which he or she was born, and the make of the first car he or she drove. It then generates and outputs their

Star Wars first and last name as well as his or her Star Wars planet.

\*/

import java.util.Scanner;

import static java.lang.System.out;

public class StarWarsName {

private static final Scanner INPUT = new Scanner(System.in);

public static void main(String args[]) {

out.println("Lab 4 written by Grecia Alieth Gonzalez");

out.println();

out.println("Star Wars Name Generator");

String firstName = promptString("Enter your first name: ");

String lastName = promptString("Enter your last name: ");

String maiden = promptString("Enter your mother's maiden name: ");

String city = promptString("Enter the city in which you were born: ");

String car = promptString("Enter the make of the first car you drove: ");

//the first 3 characters of the user's real first name, concatenated with the

//first 2 characters of the user's real last name.

while(firstName.length() < 3)

firstName += firstName;

while(lastName.length() < 2)

lastName += lastName;

// the first 2 characters of the user's mother's madien name,

//concatenated with the first 3 characters of the user's city of birth.

while(maiden.length() < 2)

maiden += maiden;

while(city.length() < 3)

city += city;

// the first 2 characters of the user's mother's madien name,

//concatenated with the first 3 characters of the user's city of birth.

String swFirstName = (firstName.substring(0,3) + lastName.substring(0,2)).toUpperCase();

String swLastName = (maiden.substring(0,2) + city.substring(0,3)).toUpperCase();

String swPlanet = (lastName.substring(lastName.length() - 2, lastName.length()) + car).toUpperCase();

out.println("You are " + swFirstName + " " + swLastName + " of " + swPlanet);

}

//This method prompts the user with a given prompt and returns his or her input (String).

private static String promptString(String prompt) {

out.print(prompt);

return INPUT.nextLine().trim();

}

}

**LAB 5**

**LAB 6**

**LAB 7**

**LAB 8**

**LAB 9**