Lab Solutions

If solutions are not available we are teaching a class and they are hidden so students don't check them until complete.

- L01
- L02
- L03
- L04 - Part 1
- L04 - Part 2
- L05
- L06
- L07
- L08
- L09
- L10
- L11
- L12

L01

Test.java

```
/**
 * Print the Answer to Life, the Universe, Everything.
 *
 * @author Brandon
 * @version 1/1/1990
 */
public class Test {
    static int answer1 = 0;
    static int answer2;
    static int theAnswer;

    public static void main(String[] args) {
        answer1 = 20;
        answer2 = 2 * answer1;
        theAnswer = answer2 + 2;
        System.out.println("The answer is...");
        System.out.println(theAnswer);
    }
}
```

ComputeCA.java

```
public class ComputeCA {
    public static void main(String[] args) {
        double radius = 4.5;
        double circumference = 22.0 / 7.0 * radius;
        double area = 22.0 / 7.0 * radius * radius;

        System.out.println("Circumference: " + circumference);
        System.out.println("Area: " + area);
    }
```

```
}
```

L02

SodaCan.java

```
/**
 * Constructor and methods for a soda can object
 *
 * @author Brandon
 * @version 1/1/1990
 */
public class SodaCan {
  private double r; // soda can radius
  private double h; // soda can height

  public SodaCan(double radius, double height) {
    r = radius;
    h = height;
  }

  public double findSurfaceArea() {
    return 2.0 * (22.0 / 7.0) * r * h + 2.0 * (22.0 / 7.0) * r * r; // compute the surface area A=2πrh+2πr^2
  }

  public double findVolume() {
    return (22.0 / 7.0) * r * r * h; // compute the volume V=πr^2h
  }
}
```

SodaCanTest.java

```
import java.util.Scanner;

/**
 * A class to test a SodaCan
 *
```

```
 * @author Brandon
 * @version 1/1/1990
 */
public class SodaCanTest {
    public static void main(String[] args) {
        Scanner myScanner = new Scanner(System.in);

        System.out.print("Enter the radius: ");
        String answer = myScanner.nextLine();
        double radius = Double.parseDouble(answer); // convert using the parseDouble method

        System.out.print("Enter the height: ");
        answer = myScanner.nextLine();
        double height = Double.parseDouble(answer); // convert using the parseDouble method

        // instantiate a soda can object using the radius and height above
        // display the surface area to three decimal digits using printf
        // display the volume to three decimal digits using printf
        SodaCan mySodaCan = new SodaCan(radius, height);

        System.out.printf("The surface area of the can is %.3f", mySodaCan.findSurfaceArea());
        System.out.println(); // this is to add a new line between the strings
        System.out.printf("The volume of the can is %.3f", mySodaCan.findVolume());
    }
}
```

L03

Coupons.java

```
import java.util.Scanner;

public class Coupons {
    public static void main(String[] args) {
        Scanner myScanner = new Scanner(System.in);
        System.out.print("Please enter the cost of your groceries: ");

        double amount = Double.parseDouble(myScanner.nextLine());

        System.out.print("You win a discount of $");
        if (amount < 10)
            System.out.print("0.");
        else if (amount <= 60)
            System.out.printf("%.2f. (8%% of your purchase)", .08 * amount);
        else if (amount <= 150)
            System.out.printf("%.2f. (10%% of your purchase)", .1 * amount);
        else
            System.out.printf("%.2f. (14%% of your purchase)", .14 * amount);
    }
}
```

L04 - Part 1

Patterns.java

```
public class Patterns {
    public static void main(String[] args) {
        for (int i = 0; i < 7; i++) {
            for (int j = 0; j < 7; j++) {
                if ((i + j) % 2 == 0)
                    System.out.print("X");
                else
                    System.out.print("_");
            }
            System.out.println(); // this is to move down a line after a row
        }
    }
}
```

L04 - Part 2

Loops.java

```
public class Loops {
    public static void main(String[] args) {
        // sum the numbers from 1 to 5 using a for loop
        int sum = 0;
        for (int i = 0; i <= 5; i++) {
            sum = sum + i;
        }
        System.out.println("The sum of the numbers from 1 to 5 is " + sum);

        // sum the numbers from 1 to 10 using a while loop
        sum = 0;
        int i = 0;
        while (i < 10) {
            i = i + 1;
            sum = sum + i;
        }
        System.out.println("The sum of the numbers from 1 to 10 is " + sum);

        // sum the numbers from 1 to 15 using a do...while loop
        sum = 0;
        i = 0;
        do {
            i = i + 1;
            sum = sum + i;
        } while (i < 15);
        System.out.println("The sum of the numbers from 1 to 15 is " + sum);
    }
}
```

L05

UsingArrays.java

```
public class UsingArrays {
    public static void main(String[] args) {
        int[] arr = { 3, 7, 13, 18 };

        for (int i = 0; i < arr.length; i++)
            System.out.println(arr[i] + " squared is " + arr[i] * arr[i]);

        for (int i = 0; i < arr.length; i++) {
            if (arr[i] < 10)
                System.out.println(arr[i] + " is less than 10");
            else if (arr[i] <= 15)
                System.out.println(arr[i] + " is greater than 10 and less than 15");
            else
                System.out.println(arr[i] + " is greater than 15");
        }
    }
}
```

ArrayTest.java

```
public class ArrayTest {
    public static void main(String[] args) {
        int[] arr = { 34, 2, 14, 34, 58, 4 };

        for (int i = 0; i < arr.length; i++)
            System.out.println(arr[i]);

        int max = 0;
        for (int i = 0; i < arr.length; i++)
            if (max < arr[i])
```

```
            max = arr[i];
        System.out.println("The max is " + max);

        int sum = 0;
        for (int i = 0; i < arr.length; i++)
            sum = sum + arr[i];
        System.out.println("The average is " + sum / arr.length);

        int[] newArr = new int[3];

        newArr[0] = arr[1];
        newArr[1] = arr[2];
        newArr[2] = arr[3];

        for (int i = 0; i < newArr.length; i++)
            System.out.println(newArr[i]);
    }
}
```

L06

BankAccount.java

```
/**
 * Parent BankAccount class
 *
 * @author Brandon
 * @version 1/1/1990
 */
public class BankAccount {
    private double balance;

    public BankAccount() {
        balance = 0;
    }

    public BankAccount(double initialBalance) {
        balance = initialBalance;
    }

    public void deposit(double amount) {
        balance = balance + amount;
    }

    public void withdraw(double amount) {
        balance = balance - amount;
    }

    public double getBalance() {
        return balance;
    }
}
```

CheckingAccount.java

```
/**
 * Child CheckingAccount class
 *
 * @author Brandon
 * @version 1/1/1990
 */
public class CheckingAccount extends BankAccount {
  private int transactionCount;

  public CheckingAccount() {
    transactionCount = 0;
  }

  public CheckingAccount(double initialBalance) {
    super(initialBalance);
    // use the super class constructor that's already coded
    transactionCount = 0;
  }

  public void deposit(double amount) {
    super.deposit(amount);
    transactionCount = transactionCount + 1;
  }

  public void withdraw(double amount) {
    super.withdraw(amount);
    transactionCount = transactionCount + 1;
  }

  public void deductFees() {
      if (transactionCount > 3) {
        double fee = 2.0*(transactionCount – 3);
        super.withdraw(fee);
      }
      transactionCount = 0;
    }
}
```

BankAccountTest.java

```
import java.util.*;

/**
 * BankAccount test class
 *
 * @author Brandon
 * @version 1/1/1990
 */
public class BankAccountTest {
    public static void main(String[] args) {
        Scanner myScanner = new Scanner(System.in);
        System.out.print("Enter initial checking account balance: ");
        double amount = Double.parseDouble(myScanner.nextLine());
        CheckingAccount myChecking = new CheckingAccount(amount);

        System.out.println("D - Deposit into checking");
        System.out.println("W - Withdraw from checking");
        System.out.println("P - End of month processing");
        System.out.println("S - Show account balance");
        System.out.println("E - Exit the program");

        boolean done = false;

        while (!done) {
            System.out.print("> ");
            String choice = myScanner.nextLine();

            if (choice.equalsIgnoreCase("D")) {
                System.out.print("Enter amount to deposit: ");
                amount = Double.parseDouble(myScanner.nextLine());
                myChecking.deposit(amount);
            } else if (choice.equalsIgnoreCase("W")) {
                System.out.print("Enter amount to withdraw: ");
                amount = Double.parseDouble(myScanner.nextLine());
                myChecking.withdraw(amount);
            } else if (choice.equalsIgnoreCase("P"))
                myChecking.deductFees();
            else if (choice.equalsIgnoreCase("S"))
                System.out.println("Checking balance: " + myChecking.getBalance());
            else if (choice.equalsIgnoreCase("E"))
```

```
            done = true;
        else
            System.out.println("Invalid menu choice - please re-enter");
    }

    System.out.println("Goodbye!");
  }
}
```

L07

Shape.java

```
public abstract class Shape {
    private String shapeName;

    public Shape() {
        shapeName = "Generic Shape";
    }

    public Shape(String shapeName) {
        this.shapeName = shapeName;
    }

    public String getShapeName() {
        return shapeName;
    }

    public abstract double findPerimeter();

    public abstract double findArea();

    public boolean equals(Object other) {
        if (other instanceof Shape) {
            Shape temp = (Shape) other;
            return this.shapeName.equals(temp.shapeName);
        }
        return false;
    }

    public String toString() {
        return "Shape Name: " + getShapeName();
    }
}
```

Circle.java

```
public class Circle extends Shape {
    private double radius;

    public Circle(double radius) {
        super("Circle");
        this.radius = radius;
    }

    public double findPerimeter() {
        return 2 * Math.PI * radius;
    }

    public double findArea() {
        return Math.PI * radius * radius;
    }

    public boolean equals(Object other) {
        if (other instanceof Circle) {
            Circle temp = (Circle) other;
            return super.equals(temp) &&
                    (this.radius == temp.radius);
        }
        return false;
    }

    public String toString() {
        return super.toString() + " radius: " + radius;
    }
}
```

Ellipse.java

```
public class Ellipse extends Shape {
    private double majorAxis, minorAxis;

    public Ellipse(double majorAxis, double minorAxis) {
```

```
        super("Ellipse");
        this.majorAxis = majorAxis;
        this.minorAxis = minorAxis;
    }

    public double findPerimeter() {
        return 2 * Math.PI * Math.sqrt(((Math.pow(majorAxis, 2) + Math.pow(minorAxis, 2)) / 2));
    }

    public double findArea() {
        return Math.PI * majorAxis * minorAxis;
    }

    public boolean equals(Object other) {
        if (other instanceof Ellipse) {
            Ellipse temp = (Ellipse) other;
            return super.equals(temp) && (this.majorAxis == temp.majorAxis) && (this.minorAxis ==
temp.minorAxis);
        }
        return false;
    }

    public String toString() {
        return super.toString() + " major axis: " + majorAxis + " minor axis: " + minorAxis;
    }
}
```

Rectangle.java

```
public class Rectangle extends Shape {
    private double length, width;

    public Rectangle(double length, double width) {
        super("Rectangle");
        this.length = length;
        this.width = width;
    }
```

```
    public double findPerimeter() {
        return 2 * length + 2 * width;
    }

    public double findArea() {
        return length * width;
    }

    public boolean equals(Object other) {
        if (other instanceof Rectangle) {
            Rectangle temp = (Rectangle) other;
            return super.equals(temp) && (this.length == temp.length) && (this.width == temp.width);
        }
        return false;
    }

    public String toString() {
        return super.toString() + " length: " + length + " width: " + width;
    }
}
```

ShapeTest.java

```
public class ShapeTest {
    public static void main(String[] args) {
        Shape[] myShapes = new Shape[10];
        Circle myCircle = new Circle(10.0);
        double expected = 314.16;
        System.out.println("Circle Area: " + myCircle.findArea());

        myShapes[0] = new Circle(10.0);
        expected = 314.16;
        System.out.println("Circle in Shapes Array Area: " + myShapes[0].findArea());

        myShapes[1] = new Ellipse(5.0, 10.0);
        expected = 157.08;
        System.out.println("Ellipse in Shapes Array Area: " + myShapes[1].findArea());

        myShapes[2] = new Rectangle(5.0, 10.0);
```

```
        expected = 50.0;
        System.out.println("Rectangle in Shapes Array Area: " + myShapes[2].findArea());
    }
}
```

L08

AutoPilot.java

```
public interface Autopilot {
    final double MAX_AIRSPEED = 550;
    double airspeed = 0;
    double altitude = 0;
    double heading = 0;

    public void setAirspeed(double inAirspeed);

    public void setAltitude(double inAltitude);

    public void setHeading(double inHeading);
}
```

Airliner.java

```
public class Airliner implements Autopilot {
    private double maxAltitude;
    double airspeed = 0;
    double altitude = 0;
    double heading = 0;

    public Airliner(double inMaxAltitude) {
        maxAltitude = inMaxAltitude;
    }

    public void setAirspeed(double inAirspeed) {
        airspeed = inAirspeed;
    }

    public void setAltitude(double inAltitude) {
        altitude = inAltitude;
```

```
    }

    public void setHeading(double inHeading) {
        heading = inHeading;
    }
}
```

L09

ReverseIt.java

```
/**
* Reverses each line of an input file and writes it to an output file.
*
* @author Brandon
* @version 1/1/1990
*/
import java.util.*;
import java.io.*;

public class ReverseIt {
    public static void main(String[] args) throws IOException, FileNotFoundException {
        Scanner myScanner = new Scanner(System.in);
        System.out.print("Enter input filename: ");
        String inFilename = myScanner.nextLine();
        System.out.print("Enter output filename: ");
        String outFilename = myScanner.nextLine();
        FileReader inReader = new FileReader(inFilename);
        PrintWriter outWriter = new PrintWriter(outFilename);

        Scanner inputFile = new Scanner(inReader);
        while (inputFile.hasNext()) {
            String inString = inputFile.nextLine();
            String outString = "";
            for (int i = 0; i < inString.length(); i++)
                outString = inString.charAt(i) + outString;
            outWriter.println(outString);
        }
        inReader.close();
        outWriter.close();
        System.out.println("Goodbye!");
    }
```

```
}
```

L10

ExceptionsLab.java

```
import java.util.Scanner;

public class ExceptionLab {
    public  static void main(String[] args) {
        int[] myInts = new int[4];
        Scanner scan = new Scanner(System.in);

        try {
            for (int i = 0; i != 4; i++) {
                System.out.println("Enter a integer: ");
                int num = Integer.parseInt(scan.nextLine());
                myInts[i] = num;
            }

            System.out.println("Your Numbers");

            for (int i = 0; i != 4; i++) {
                System.out.print(myInts[i] + " ");
            }
        }
        catch (NumberFormatException myEx){
            System.out.println("You should have entered a Integer you TWAT!");
        }
    }
}
```

L11

InsufficientFunds.java

```
public class InsufficientFunds extends RuntimeException{
    public InsufficientFunds(){

    }
    public InsufficientFunds(String msg){
        super(msg); // use superclass constructor
    }
}
```

BankAccount.java

```
public class BankAccount{
  private double balance;
  public BankAccount(){
    balance = 0;
  }
  public BankAccount(double initialBalance){
    balance = initialBalance;
  }
  public void deposit(double amount){
    balance = balance + amount;
  }
  public void withdraw(double amount){
    if(amount <= balance)
      balance = balance - amount;
    else{
      InsufficientFunds myEx = new InsufficientFunds("Amount Exceeds Balance");
      throw myEx;
    }
  }
  public double getBalance(){
```

```
    return balance;
  }
}
```

CheckingAccount.java

```
public class CheckingAccount extends BankAccount
{
  private int transactionCount;
  public CheckingAccount(){
    transactionCount = 0;
  }
  public CheckingAccount(double initialBalance){
    super(initialBalance); //use the super class constructor that's already coded
    transactionCount = 0;
  }
  public void deposit(double amount){
    super.deposit(amount);
    transactionCount = transactionCount + 1;
  }
  public void withdraw(double amount){
    super.withdraw(amount);
    transactionCount = transactionCount + 1;
  }
  public void deductFees(){
    if (transactionCount > 3){
      double fee = 2.0*(transactionCount - 3);
      super.withdraw(fee);
    }
    transactionCount = 0;
  }
}
```

BankAccountTest.java

```
import java.util.*;


public class BankAccountTest
```

```
{
  public static void main(String[] args)
  {
    Scanner myScanner = new Scanner(System.in);
    System.out.print("Enter initial checking account balance: ");
    double amount = Double.parseDouble(myScanner.nextLine());
    CheckingAccount myChecking = new CheckingAccount(amount);
    System.out.println("D - Deposit into checking");
    System.out.println("W - Withdraw from checking");
    System.out.println("P - End of month processing");
    System.out.println("S - Show account balance");
    System.out.println("E - Exit the program");
    boolean done = false;
    while (!done)
    {
      System.out.print("> ");
      String choice = myScanner.nextLine();
      if (choice.equalsIgnoreCase("D"))
      {
        System.out.print("Enter amount to deposit: ");
        amount = Double.parseDouble(myScanner.nextLine());
        myChecking.deposit(amount);
      }
      else if (choice.equalsIgnoreCase("W")) {
        try {
          System.out.print("Enter amount to withdraw: ");
          amount = Double.parseDouble(myScanner.nextLine());
          myChecking.withdraw(amount);
        } catch (InsufficientFunds myEx) {
            System.out.println(myEx.getMessage());
          System.out.println("Transaction Aborted!");
        }
      }
      else if (choice.equalsIgnoreCase("P"))
        myChecking.deductFees();
      else if (choice.equalsIgnoreCase("S"))
        System.out.println("Checking balance: " + myChecking.getBalance());
      else if (choice.equalsIgnoreCase("E"))
        done = true;
      else
```

```
        System.out.println("Invalid menu choice - please re-enter");
    }
    System.out.println("Goodbye!");
  }
}
```

L12

FibonacciTest.java

```
import java.util.*;

public class FibonacciTest {
    public static void main(String[] args) {
        Scanner myScanner = new Scanner(System.in);
        boolean done = false;
        while (!done) {
            System.out.print("Enter an integer or Q to quit: ");
            String answer = myScanner.nextLine();
            if (answer.equalsIgnoreCase("q"))
                done = true;
            else {
                int myLong = Integer.parseInt(answer);
                long startTime = System.currentTimeMillis();
                long fibN = fibonacci(myLong);
                long endTime = System.currentTimeMillis();
                double elapsedTime = (endTime - startTime) / 1000.0;
                System.out.print("Fibonacci(" + myLong + ") = " + fibN);
                System.out.println(" took " + elapsedTime + " seconds");
            }
        }
        System.out.println("Goodbye!");
    }

    public static long fibonacci(long n) {
        if (n <= 2) {
            if (n == 0)
                return (0);
            else
                return (1);
        } else
            return (fibonacci(n - 1) + fibonacci(n - 2));
```

```
    }
}
```

ExponentTest.java

```
import java.util.Scanner;

public class ExponentTest {
    public static void main(String[] args) {
        Scanner myScanner = new Scanner(System.in);
        boolean done = false;
        while (!done) {
            System.out.print("Enter an integer a then n to compute a^n or Q to quit: ");
            String answer = myScanner.nextLine();
            if (answer.equalsIgnoreCase("q"))
                done = true;
            else {
                int a = Integer.parseInt(answer);
                System.out.print("Enter an integer n: ");
                answer = myScanner.nextLine();
                int n = Integer.parseInt(answer);
                long startTime = System.currentTimeMillis();
                long exp = exponent(a, n);
                long endTime = System.currentTimeMillis();
                double elapsedTime = (endTime - startTime) / 1000.0;
                System.out.print("exponent(" + a + ", " + n + ") = " + exp);
                System.out.println(" took " + elapsedTime + " seconds");
            }
        }
        System.out.println("Goodbye!");
    }

    public static int exponent(int a, int n) {
        if (n > 1) {
            return a * exponent(a, n - 1);
        } else {
            return a;
        }
```

```
    }
}
```

Reverse.java

```
import java.util.Scanner;

public class Reverse {
    public static void main(String[] args) {
        Scanner myScanner = new Scanner(System.in);
        boolean done = false;
        while (!done) {
            System.out.print("Enter a string to reverse or Q to quit: ");
            String answer = myScanner.nextLine();
            if (answer.equalsIgnoreCase("q"))
                done = true;
            else {
                long startTime = System.currentTimeMillis();
                String rev = reverse(answer);
                long endTime = System.currentTimeMillis();
                double elapsedTime = (endTime - startTime) / 1000.0;
                System.out.print("reverse(" + answer + ") = " + rev);
                System.out.println(" took " + elapsedTime + " seconds");
            }
        }
        System.out.println("Goodbye!");
    }

    public static String reverse(String s) {
        if (s.length() > 0)
            return s.charAt(s.length() - 1) + reverse(s.substring(0, s.length() - 1));
        else
            return "";
    }
}
```