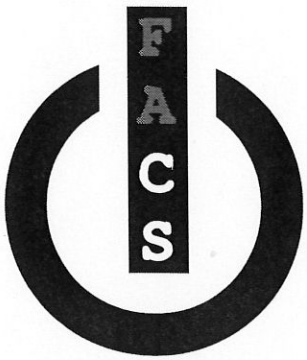
Computer Science 2
Introduction to Java
Fort Atkinson High School



turnOnThePower

Computer Science 2 Curriculum Outline

1. **Console Programming**
   - Unit 1: Displaying simple output
   - Unit 2: Input-Process-Output apps
   - Unit 3: Decision Making
   - Unit 4: Random Numbers
   - Unit 5: The 'for' loop
   - Unit 6: Accumulation
   - Unit 7: The 'while' loop
   - Unit 8: Arrays
   - Unit 9: ArrayLists

2. **GUI Programming**
   - Unit 1: Basic input and output using Dialog windows
   - Unit 2: Drop down menus using Customized dialog
   - Unit 3: GUI based menu driven programs

3. **Game Programming**
   - Unit 1: Simple Demo Project
   - Unit 2: Tetris Project

4. **NetBeans Gui Builder**
   - Unit 1: Frames and Panels
   - Unit 2: Labels and TextFields
   - Unit 3: Buttons and Events

Computer Science 2
Console Programming
Unit 1: Displaying simple output

Things to learn while doing this assignment:

The statement System.out.println("insert words here") is used to print to the console screen.
The statement System.out.println(); issues a new line
The statement System.out.print("insert words here") does not issue a carriage return
The syntax for java is very picky. You must pay attention to details!

Required Programs (You must do each of these!)
1. Poem
2. Tell me about yourself
3. Name printer
4. Print vs println
5. Fix me
6. Extra program of your design and creativity

1) **Program: Poem**
   Class name: Poem
   Write a program that prints this poem:

   Roses are red,
   Violets are blue,
   Binary is sweet,
   And so are you.

2) **Program: Tell me about yourself**
   Class name: TellMeAboutYourself

   Write a program that prints a short two (2) paragraph autobiography.

   Example:
   Hello, I'm DrJ. I'm not really a doctor, my initials are D.R.J. I have loved computer programming since I was a senior in high school.

   I like to work in my workshop where I've made furniture for my home. I also have gone kayaking and camping around the Apostle Islands.

3) **Program: Name printer**
Class name: NamePrinter

Write a program that displays your name inside a box on the console screen, like this:

```
+---------+
|  Dr. J  |
+---------+
```

4) **Program: Print vs println**
Class name: PrintVsPrintln

Predict the output for this program. Then, copy and paste it into DrJava. Compile and run the program and compare the result to your prediction.

```
public class PrintVsPrintln
{
   public static void main(String[] args)
   {
      System.out.print("Hello,");
      System.out.print(" how");
      System.out.print(" are you");
      System.out.println("?");
   }
}
```

5) **Program: Fix me**
Class name: FixMe

The following program contains many syntax errors. Copy and paste it into DrJava. Compile, debug until you are able to run the program.

```
public class FixMe
{
   Public static void Main(string args)
   {
      system.out.print(Help me,")
      System.out.printLn(I'm broken!);
}
```

Computer Science 2 **Name** _____
Unit 1 Print vs Println **Quiz #1 A high-tech world**

```
public class PrintQuiz1
{
  public static void main(String[] args)
  {
    // State the output of this program on the lines below
    System.out.println("A high-tech world");
    System.out.print("calls for inspired ");
    System.out.println("and talented people.");
    System.out.println();
    System.out.print("Jobs in technology ");
    System.out.print("are growing ");
    System.out.println("and ");
    System.out.println("offer ");
    System.out.println("big ");
    System.out.println();
    System.out.println("salaries.  ");
    System.out.println("Technological advances are ");
    System.out.print("creating ");
    System.out.println("new ");
    System.out.println("jobs every day!");
  }
}
```

Computer Science 2 **Name** _____
Unit 1 Print vs Println **Quiz #2 From entertainment**

```
public class PrintQuiz2
{
  public static void main(String[] args)
  {
    // State the output of this program on the lines below
    System.out.print("From entertainment ");
    System.out.println("and communication ");
    System.out.println("to ");
    System.out.print("conquering disease or ");
    System.out.println();
    System.out.print("eliminating ");
    System.out.println("hunger - computer ");
    System.out.print("technologies ");
    System.out.println("provide ");
    System.out.println("solutions ");
    System.out.println("to make ");
    System.out.println();
    System.out.println("life ");
    System.out.println("better.");  }
}
```

Computer Science 2
Console Programming
Unit 2: Input-Process-Output

Things to learn while writing these programs:

Integer numbers should be stored in 'int' variables
Decimal numbers should be stored in 'double' variables
Word values should be stored in 'String' variables
You must think ahead of time about what will be stored in a variable before you create it. This helps you declare the right 'type' of variable (int, double, String, etc)
All variables start with a lower case letter. They cannot start with a symbol or number.
You can print variables by putting them in the System.out.println(insert variable here);
Input – Process – Output style programs are used to create simple apps
You can use the Scanner class to get information from the user using the keyboard
You need the import statement: import java.util.Scanner; to use the Scanner class
You cannot use the same Scanner object for both numbers and Strings. You need to create one Scanner object for all numbers, and another Scanner object for String input
An algorithm is a sequence of steps that when followed perform a specific task.
Programmers develop algorithms before typing programs.

Required Programs (You must do each of these!)
1. Miles per gallon calculator
2. Celsius to Fahrenheit converter
3. Stadium seating
4. Kilometer converter
5. Mad lib
6. Extra program using your own idea

1) **Program: Miles per gallon calculator**
Class name: MPGCalculator
Design and implement a program that computes the miles per gallon for a car. Ask the user to enter a distance in miles and a number of gallons and then computes and displays the miles per gallon. Always test your programs with data to see if it works with reasonable data. (Hint: The miles traveled should be an integer, the number of gallons should be a double, and the miles per gallon should be a double.)

Example:
Enter the number of miles traveled: 100
Enter the number of gallons used: 4.8

Your miles per gallon are 20.8333333.

2) **Program: Celsius to Fahrenheit converter**
Class name: CelsiusToFahrenheitConverter
Design and implement a program that converts Celsius to Fahrenheit. Ask the user to enter a Celsius temperature and then compute and display the Fahrenheit temperature. Hint: The formula for conversion is F = (9/5)C + 32.

Example:
Please enter a Celsius temperature: 25
25 degrees Celsius is equal to 77 degrees Fahrenheit

3) **Program: Stadium Seating**
Class name: StadiumSeating
There are three seating categories at a stadium. For a game, Class A seats cost $15, Class B seats cost $12, and Class C seats cost $9. Design and implement a program that asks how many tickets for each class of seats were sold, and then display the amount of income generated from the ticket sales for each class and the total income for the game.

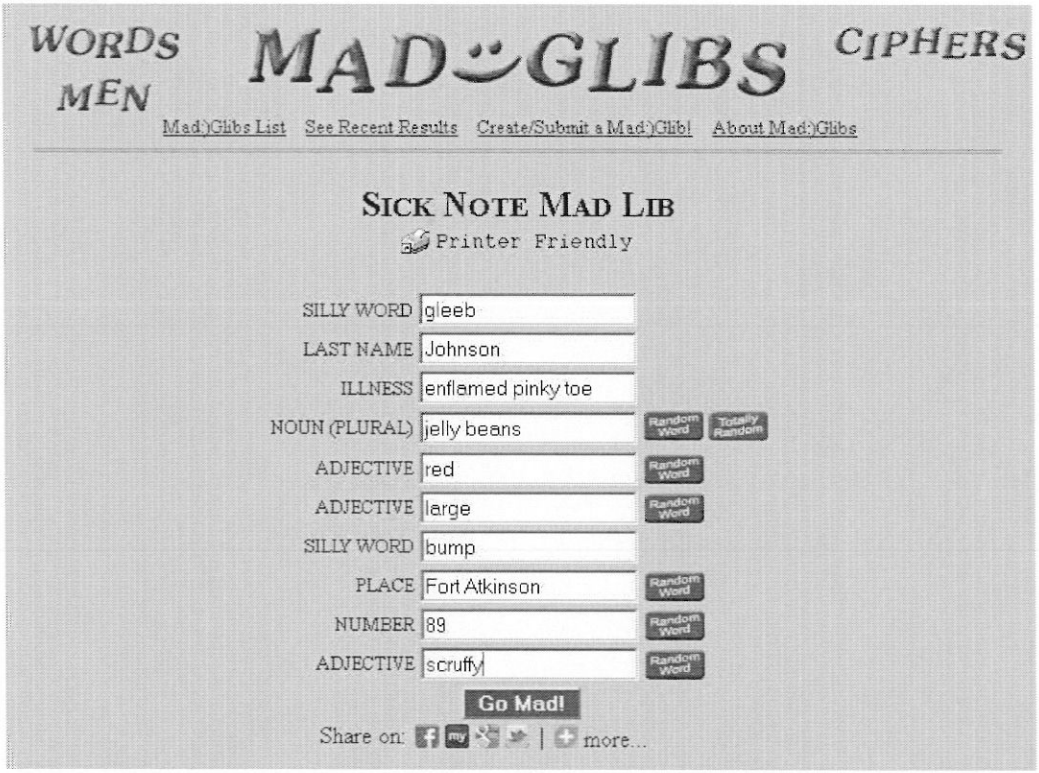
Example:
Enter the number of A tickets sold: 100
Enter the number of B tickets sold: 150
Enter the number of C tickets sold: 275
The total income from these ticket sales is $5775.

4) **Program: Kilometer converter**
Class name: KilometerConverter
Design and implement a program that converts kilometers to miles. Ask the user to enter a distance in kilometers, and then converts and displays that distance to miles. To be a friendly program, ask the user to enter their name and then use their name when communicating with them. (The conversion formula is: miles = kilometers x 0.6214)

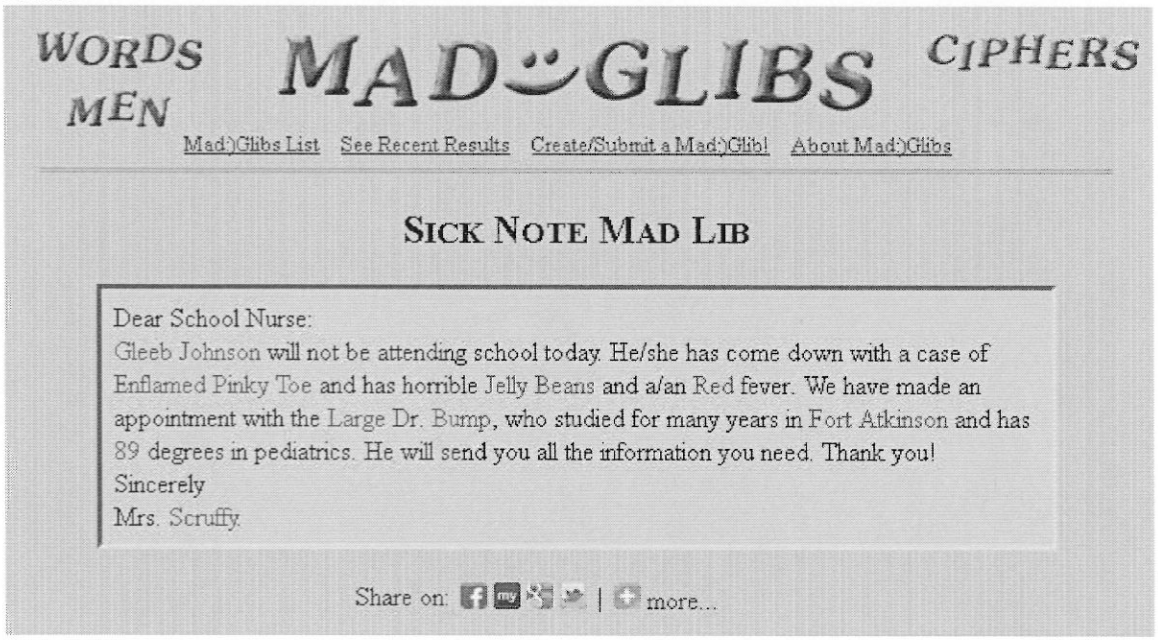
Example:
Enter your name: DoctaJ
DoctaJ, please enter a distance in kilometers: 10
DoctaJ, this distance in miles is 6.214.

5) **Program: Mad Lib**
Class name: MadLib
Create a Mad Lib of the similar format at the following example. Create a separate String variable for each required word.

Before:



After:



Computer Science 2 **Name** _____
Unit 2 Input-Process-Output **Quiz #1 Simple Interest**

Write a program to compute the interest earned in a savings account. The formula to calculate the interest is given by: Interest earned = Principal x Rate x Time. You must use appropriate names when declaring all variables. The interest rate is the only input that is allowed to be a decimal number.

The program MUST run exactly like this:

Welcome to the Interest Calculator!
Enter your name: Superman
Enter the amount of the principal: 1000
Enter the interest rate: .07
Enter the number of years: 5

Superman, the amount of interest you would earn is $350.0

```
import java.util.Scanner;
public class SimpleInterest
{
  public static void main(String[] args)
  {
    Scanner wordInput = new Scanner(System.in);
    Scanner numberInput = new Scanner(System.in);









  }
}
```

Computer Science 2 **Name** _____
Unit 2 Input-Process-Output **Quiz #2 BMI**

The Body Mass Index (BMI) is often used to determine whether a person with a sedentary lifestyle is overweight or underweight for their height. A person's BMI is calculated with the formula below. Write a program that calculates the BMI given a weight (in pounds) and height (in inches). The weight can be a decimal number. The height cannot be a decimal number.

$$BMI = Weight \times \frac{703}{Height^2}$$

The program MUST run exactly like this:
Welcome to the Body Mass Index Calculator!
Enter your name: Barbie
Enter your weight: 110.3
Enter your height: 68

Barbie, your BMI is 16.7692257785

```
import java.util.Scanner;
public class BMI
{
  public static void main(String[] args)
  {
    Scanner wordInput = new Scanner(System.in);
    Scanner numberInput = new Scanner(System.in);

  }
}
```

Computer Science 2
Console Programming
Unit 3: Decision Making (if-else)

Things to learn while writing these programs:

An 'if-else' statement is referred to as a conditional statement
The act of decision making is made using 'if-else' statements
You can use either an 'if' or an 'if-else'
Conditional statements can be nested (one can be put inside of another)
Conditional statements use the comparators: <, <=, >, >=, == (equals), ! (not)
In order to use the word AND, you must replace it with && (two ampersands)
In order to use the word OR, you must replace it with || (two vertical bars)
String variables can only be compared using the .equals method. Example: String name1 = "Joe"; String name2 = "Joe"; if (name1.equals(name2))   System.out.println("hey, we have the same name!");
String comparisons are case sensitive (unless you use the method that ignores the case)

Required Programs (You must do each of these!)
1. Book Club Points
2. Shipping Charges
3. Software Sales
4. Username and Password
5. Letter Grade Comment
6. Extra program of your own design

1. **Program: Book Club Points**
   Class name: BookClubPoints
   ComicSans Booksellers has a book club that awards points to its customers based on the number of books purchased each month. The points are awarded as follows:
   - If a customer purchases 0 books, then he or she earns 0 points.
   - If a customer purchases 1 book, then he or she earns 5 points.
   - If a customer purchases 2 books, then he or she earns 10 points.
   - If a customer purchases 3 books, then he or she earns 30 points.
   - If a customer purchases 4 or more books, then he or she earns 60 points.

Design and implement a program that asks the user to enter the number of books that he or she has purchased this month and displays the number of points awarded.

Example:
Please enter the number of books purchased this month: 3
You get 30 award points!

2. **Program: Shipping Charges**
   Class name: ShippingCharges
   The Fast Freight Shipping Company charges the following rates:

Weight of Package	Rate per Pound
2 pounds or less	$1.10
Over 2 pounds but not more than 6 pounds	$2.20
Over 6 pounds but not more than 10 pounds	$3.70
Over 10 pounds	$3.80

Design and implement a program that asks the user to enter the weight of a package and then displays the shipping charges for that package.

Example:
Please enter the weight of the package: 14
The shipping charge is $3.80

3. **Program: Software Sales**
   Class name: SoftwareSales
   A software company sells a package that retails for $99. Quantity discounts are given according to the following table:

Quantity	Discount
0 - 9	0%
10 – 19	20%
20 – 49	30%
50 – 99	40%
100 or more	50%

Design and implement a program that asks the user to enter the number of packages purchased. The program should then display the percent of the discount, the cost prior to the discount, the amount of the discount (if any) and the total amount of the purchase after the discount.

Example:
Please enter the number of packages that you bought: 25
You earn a 30% discount.
The total before the discount is $2475.
The discount is $742.0
The final cost is $1733

4. **Program: Username and Password**
   Class name: UsernameAndPassword
   Usernames and passwords are used every day by people around the world. How does the computer know when the user has entered the correct values?
   Design and implement a program that allows the user to 'login' to the computer with a username and password. If they get both correct, then the computer displays a message that says, "Logging in…". If they don't enter both of them correctly, then display a message that says, "Login failed. Username or password is incorrect."

   Example:
   Enter Username: Zaspelj
   Enter Password: csrocks

   Logging in…

   (If run again with incorrect values)
   Example:
   Enter Username: Zaspelj
   Enter Password: sciencerocks

   Login failed. Username or password is incorrect.

5. **Program: Letter Grade Comment**
   Class name: LetterGradeComment
   Letter grades can be used to assign a comment on a grading program. Comments are given according to the following table:

Letter Grade	Comment
A	Excellent work!
B	Good work
C	Average work
D	Below average work
F	Failing work

   Design and implement a program that asks the user to enter the letter grade for a student. The program should then display the appropriate comment for that letter grade.

   Example:
   Please enter the letter grade for a student: A
   Comment: Excellent work!

Computer Science 2 **Name** _____
Unit 3 Decision Making **Quiz #1 First Class Envelope**

The United States Postal Service charges the following rates for a first class envelope:

Weight of Envelope	Stamp Price
1 ounce or less	$0.45
Over 1 ounce but not over 2 ounces	$0.65
Over 2 ounces but not over 3 ounces	$0.85
Over 3 ounces but not over 3.5 ounces	$1.05

Design and implement a program that asks the user to enter the weight of a first class envelope and then displays the stamp price.

The program MUST run exactly like this:
Please enter the weight of the first class envelope: 1.2
The stamp price is $0.65.

```
import java.util.Scanner;
public class FirstClassEnvelope
{
  public static void main(String[] args)
  {
    Scanner numberInput = new Scanner(System.in);









  }
}
```

Computer Science 2 **Name** _____
Unit 3 Decision Making **Quiz #2 Overtime Pay**

Program: Overtime Pay

When an hourly wage worker works more than 40 hours in one week, they earn overtime for the hours that they work over 40. The overtime rate is one-and-one-half times their normal hourly rate. Write a program that calculates the pay for an employee if they know their hourly rate and the number of hours they worked. The hourly rate is a decimal number and the hours worked is not a decimal.

Consider the following **two** examples:

Enter the hourly wage: 8.00
Enter the hours worked: 30
Your total pay is $240.0 (Pay = $8.00*30 = $240.00)

Enter the hourly wage: 8.00
Enter the hours worked: 50
Your total pay is $440.0 (Pay = $8.00 * 40 + $12*10 = $320.00 + $120.00 = $440.00)

```
import java.util.Scanner;
public class OvertimePay
{
  public static void main(String[] args)
  {
    Scanner numberInput = new Scanner(System.in);









  }
}
```

Computer Science 2 **Name** _____
Unit 3 Decision Making **Quiz #3 Debt to Income Ratio**

Directions:
When a person is applying for a loan for a car, the loan officer looks at many of your attributes. One of these is called your *Debt-to-Income Ratio*. Here is how the ratio is computed:

$$Debt\text{-}to\text{-}Income\ Ratio = \frac{TotalMonthlyDebt}{MonthlyGrossPay}$$

The user will be asked to enter the Total Monthly Debt and the Monthly Gross Pay. The app computes the Debt-to-Income Ratio and tells the user if they need a cosigner.

This Ratio will result in a decimal value.

- If the Ratio is less than 0.36, then the borrower is approved for the loan.
- If the Ratio is greater than or equal to 0.36 but less than .4, then the borrower is approved, but only with a cosigner.
- If the Ratio is greater than or equal to .4, then the borrower is denied the loan.

Example 1
Enter the Total Monthly Debt = 800
Enter the Monthly Gross Pay = 2100 (Debt-to-Income Ratio = 800/2100= 0.38095)

Your Debt-to Income ratio is 38.0%
Result: You are approved, but you'll need a cosigner!

Example 2
Enter the Total Monthly Debt = 900
Enter the Monthly Gross Pay = 3000 (Debt-to-Income Ratio = 900/3000= 0.3)

Your Debt-to Income ratio is 30.0%
Result: Congratulations! You are approved without a cosigner!

Write the program on the back of this sheet.

Computer Science 2
Console Programming
Unit 4: Random Numbers

Things to learn while writing these programs:

Random numbers are very powerful when writing programs that do simulations
There are two ways to create random numbers in Java
One way is to use the Math.random() statement
Math.random() returns a double that is between 0 and 1. (including 0 but not including 1)
To create a random integer using Math.random(), Example: int winningNumber = (int)(Math.random()*totalPossibilities) + lowestNumber;
The second way is to use the Random class
You need to create an object from the Random class in order to generate random numbers Example: import java.util.Random; Random myGenerator = new Random(); int winningNumber = myGenerator.nextInt(maxNumber) + lowestNumber;

Required Programs (You must do each of these!)
1. Pick a number out of a hat
2. Coin flip
3. Power Ball
4. Random direction chooser
5. Extra program of your own design

1. **Program: Pick a number out of a hat**
   Class name: PickANumber
   Suppose a teacher assigns a number to everyone in her class. There are 25 students in the class. Design and implement a program that randomly chooses a number between 1 and 25. Display the number.

   Example:
   The winning number is 13.

   (If you run the program again, you get another randomly chosen number.)
   The winning number is 9.

2. **Program: Coin flip**
   Class name: CoinFlip
   Design and implement a program that simulates the flipping of a coin one time. Simulate this by having the computer pick between two random numbers, such as 0 and 1. If the random number is a 0, then display 'heads'. If the random number is a 1, then display 'tails'.

   Example:
   Heads

   (If you run the program again, you get a random result.)
   Tails

3. **Program: Power Ball**
   Class name: PowerBall

   Design and implement a program that simulates the PowerBall lottery. The program should display 6 numbers that were randomly chosen between 1 and 40. For this program, it is ok if the same ball is chosen more than once
   CHALLENGE: Make sure that the same number is not chosen more than once.

   Example:
   The Winning Numbers Are
   25 7 15 40 19 10

   (If you run the program again, you get a random result.)
   The Winning Numbers Are
   40 9 33 18 24 36

4. **Program: Random direction chooser**
   Class name: DirectionChooser

   Suppose you were working on a game that had an object move randomly around the screen. You would want the computer to randomly choose a direction between up, down, left, or right.

   Design and implement a program that simulates this random movement. The program should assign a number to each direction. Since there are 4 directions, it seems logical to assign the directions in the following manner:

   1 = up
   2 = down
   3 = left
   4 = right

Have the program pick a number randomly between 1 and 4. If the number is 1, then display 'Move up'. If the number is 2, then display 'Move down', and so on.

Example:
Move up

(If you run the program again, then the computer will choose another direction.)
Move left

Computer Science 2 **Name** _____
Unit 4 Random Numbers **Quiz #1 Pick a Student**

There are roughly 960 students at Fort Atkinson High School. Suppose that a lottery will be held to pick a winning student from the list of students. Each student in the high school will be assigned a unique number as given in the table. Design and implement a program that randomly picks a sophomore, a junior and a senior.

Freshmen: 1 - 250
Sophomores: 251 – 495
Juniors: 496 – 722
Seniors: 723 - 960

The program MUST run exactly like this (aside from the random numbers!):
The sophomore winner is 375.
The junior winner is 512.
The senior winner is 947.

```
import java.util.Scanner;
public class PickAStudent
{
  public static void main(String[] args)
  {
    Scanner numberInput = new Scanner(System.in);

  }
}
```

Computer Science 2 **Name** _____
Unit 4 Random Numbers **Quiz #2 Pick a Card**

There are 52 cards in a standard deck. There are four suits (Clubs, Spades, Hearts, and Diamonds) and 13 values (1-10, jack, queen, king). Write a program that selects a card at random and displays the card.

The program MUST run exactly like this (aside from the random card!):
Suit = clubs
Value = 12
Your card is the 12 of clubs.

```
public class PickACard
{
  public static void main(String[] args)
  {

  }
}
```

Computer Science 2
Console Programming
Unit 5: The 'for' loop

Things to learn while writing these programs:

The 'for' loop is a programming structure that allows one instruction (or a set of instructions) to be repeated a specific number of times.

The general structure looks like this:
for (int *index* = *start value*; *index* comparator *end value*; *increment index*)
{
instructions;
}

Example: To print the numbers from 3 to 8
for (int i = 3; i < 9; i++) This is the preferred way to count from 3 to 8.
system.out.println(i);
 	
or
 	
for (int i = 3; i <= 8; i++) This technique is NOT preferred due to comparisons of
system.out.println(i); exact values of equality are not always achieved.

The *index* is called the 'loop control' variable because it controls the loop
You may increment by any number that you want to.
To increment by 1, you can use ++. Example: i++ means i = i + 1 and i-- means i = i - 1
You may also decrement the loop control variable.
To repeat more than one instruction, you must put the instructions in a 'block' of code.
As with many programs, there may numerous ways to achieve the same result, however, a good programmer always seeks to find the most efficient and elegant way.

Required Programs (You must do each of these!)
1. Simple list of numbers
2. Square root table
3. Temperature Table
4. Trig Table
5. Extra program of your own design

1. **Program: Simple list of numbers**
   Class name: SimpleList

Design and implement a program that uses a 'for' loop to print the integers from 13 to 29 (inclusive). Display each number on a separate line.

Example:
13
14
15
:
:
29

2. **Program: Square Root Table**
   Class name: SquareRootTable
   Design and implement a program that uses a 'for' loop to create a table of square roots for the integers from 1 – 10. The table must be created during the running of the program. (Hard-coding the table is not allowed). Both columns must have a heading and the square root must display exactly 3 decimal places.

   Example:

Number	Square Root
1	1.000
2	1.414
3	1.732
4	2.000
…	…
10	3.162

3. **Program: Temperature Table**
   Class name: TemperatureTable
   Design and implement a program that uses a 'for' loop to create a Celsius–Fahrenheit table for the Celsius temperatures from 0 to 100 counting by 5's. The table must have headings and may not be hard-coded.
   CHALLENGE: Ask the user to input what integer to start the table at and what integer to end the table at.

   Example:

Celsius	Fahrenheit
0	32
5	41
10	50
15	59
:	:
:	:
100	212

4. Trig Table

Design and implement a program that creates a trigonometry table that displays the sine, cosine and tangent values (rounded to four decimal places) for each integer valued degree from 0 to 30 degrees. The table must have headings and may not be hard-coded. The trig values must display 4 decimal places. HINT: Remember that these trig values require the parameter to be in radian measure.

Example:

Degree	Sine	Cosine	Tangent
0	0.0000	1.0000	0.0000
1	0.0175	0.9998	0.0175
2	0.0349	0.9993	0.0349
...	...	...	...
30	0.5000	0.8860	0.5773

Computer Science 2 **Name** _____
Unit 5 The 'for' loop **Quiz #1 For loop analysis 1**

Write a program that uses a for loop to display the odd numbers from 3 to 15 as shown.

The program MUST run exactly like this:
3 5 7 9 11 13 15

```
import java.util.Scanner;
public class ForLoopAnalysis1
{
  public static void main(String[] args)
  {
    Scanner numberInput = new Scanner(System.in);









  }
}
```

Predict the output of this program.

```
public class Mystery
{
  public static void main(String[] args)
  {
    for (int i = 1; i < 5; i = i + 1)
    {
      System.out.println(i*i);
    }
  }
}
```

Computer Science 2 **Name** _____
Unit 5 The 'for' loop **Quiz #2 For loop analysis 2**

Write a program that uses a for loop to display the even numbers from 4 to 12 as shown.

The program MUST run exactly like this:

```
4
6
8
10
12
```

```
public class ForLoopAnalysis2
{
  public static void main(String[] args)
  {









  }
}
```

Predict the output of this program.

```
public class Mystery
{
  public static void main(String[] args)
  {
    for (int i = 3; i < 16; i = i + 3)
    {
      System.out.println(i);
    }
  }
}
```

_____

_____

_____

_____

_____

_____

_____

Computer Science 2
Console Programming
Unit 6: Accumulation

Things to learn while writing these programs:

The word accumulation means 'to accumulate' or add to a total
The act of accumulating occurs in programming when you add a number to a total. Example: total = total + number (Read as "total takes on the value of the old total plus number")

To use a for loop and accumulation, follow this algorithm:
1. Create a total variable and set it equal to 0
2. for (as many times as you want)
{
Get a new value
total = total + new value
}
3. Display total

Required Programs (You must do each of these!)
1. Simple accumulation
2. Golf score calculator
3. Grade book
4. Voting for class president
5. Extra program of your own design

1. **Program: Simple accumulation**
   Class name: SimpleAccumulation
   Design and implement a program that uses accumulation to add up 3 numbers entered by the user. You must add each number to the total after the number is entered. Display the current total after each number is entered. Don't use a loop.

   Example:
   Please enter the first score: 80
   Your total after one score is 80.

   Please enter the second score: 90
   Your total after two scores is 170.

   Please enter the third score: 60
   Your total after three scores is 230.

2. **Program: Golf Score Calculator**
   Class name: GolfScoreCalculator
   Design and implement a program that uses a 'for' loop and accumulation to total up 9 golf scores for the user. The program should ask the user for their golf score for each hole. The program should keep track of the total score by adding the new score to the total. After the user has entered the score for the 9th hole, the program should display the total. *YOU **MUST** USE A FOR LOOP.*

   Example:
   Enter score for hole #1: 6
   Enter score for hole #2: 5
   Enter score for hole #3: 7
   Enter score for hole #4: 4
   Enter score for hole #5: 5
   Enter score for hole #6: 8
   Enter score for hole #7: 3
   Enter score for hole #8: 5
   Enter score for hole #9: 4

   Your total for this round of golf was 47.

3. **Program: Grade book**
   Class name: Gradebook
   Design and implement a program that uses a 'for' loop and accumulation to help a teacher calculate the class average for a test. The program should ask the teacher for exactly how many scores are going to be entered and then ask the teacher for each score on the test. After the teacher has entered the final score, the program should calculate and display the average.

   Example:
   How many students took the test? 25
   Enter test score 1: 78
   Enter test score 2: 95
   Enter test score 3: 65
   Enter test score 4: 89
   :
   :
   Enter test score 25: 71

   The class average on this test is 82.1.

4. **Program: Voting for class president**
   Class name: VoteForClassPresident
   Design and implement a program that uses a 'for' loop and accumulation to determine the result of an election for class president. Alex and Danielle are the two candidates. The program will ask for how many ballots will be cast. Then, the program will prompt the user to enter either "Alex" or "Danielle". If the vote is valid, then the program will state, "Your vote counted." If the vote was inaccurate, then the program will state, "Your vote is invalid."

   After all of the votes have been cast, the program will display the results for both candidates, the number of invalid votes, and state who won the election. If the election ends in a tie, state so.

   Example:
   How many votes will be cast? 10

   Vote for Alex or Danielle: Danielle
   Your vote counted.
   Vote for Alex or Danielle: Alex
   Your vote counted.
   Vote for Alex or Danielle: Dan
   Your vote is invalid.
   Vote for Alex or Danielle: Danielle
   Your vote counted.
   :
   :
   Here are the results of the election:
   Alex had 4 votes.
   Danielle had 5 votes.
   Invalid votes = 1.

   Danielle won the election.

Computer Science 2 **Name** _____
Unit 6 Accumulation **Quiz #1 Accumulation**

Predict the output of this program. SHOW MEMORY STORAGE!

```
public class Mystery
{
  public static void main(String[] args)
  {
    int t = 0;
    for (int i = 1; i < 5; i = i + 1)
    {
      t = t + i;
    }
    System.out.println(t);
  }
}
```

Memory Storage

Predict the output of this program. SHOW MEMORY STORAGE!

```
public class Mystery
{
  public static void main(String[] args)
  {
    int t = 5;
    int a = 4;
    t = t + a;
    int b = 3;
    t = t + b;
    t = t + t;
    System.out.println(t);
  }
}
```

Memory Storage

Computer Science 2 **Name** _____
Unit 6 Accumulation **Quiz #2 Accumulation**

Predict the output of this program. SHOW MEMORY STORAGE!

```
public class Mystery
{
  public static void main(String[] args)
  {
    int t = 0;
    for (int k = 2; k < 6; k = k + 1)
    {
      t = t + k;
    }
    System.out.println(t);
  }
}
```

Memory Storage

OUTPUT

Predict the output of this program. SHOW MEMORY STORAGE!

```
public class Mystery
{
  public static void main(String[] args)
  {
    int n = 13;
    int p = 7;
    n = n + p;
    int v = 2;
    n = n + v;
    n = 3*n;
    System.out.println(n);
  }
}
```

Memory Storage

OUTPUT

Computer Science 2
Console Programming
Unit 7: The 'while' loop

Things to learn while writing these programs:

The 'while' loop is used to repeat instructions as long as some condition is true.
You read the while statement as "as long as"

The structure of the while loop is:
while (some condition is true)
{
  instructions;
}

As soon as the condition becomes false, the program exits the while loop and executes the next instruction
The programmer is responsible for making sure that the while loop will be able to end
A while loop that has no possibility of ending creates an infinite loop (a loop that never ends).

Here is the **standard algorithm** for counting and accumulating when using a while loop:
 	
Set the counter to zero.
Set the total to zero.
Ask the user for input
While (the input is valid)
  {
  Increment the counter
  Add the input to the total
  Perform other computations that are specific to the program
  Ask the user for input
  }

You can use && and || as part of any condition

Required Programs (You must do each of these!)
1. Guess my number
2. Secret pass code
3. Checkout counter
4. Darts 301
5. Extra program of your own design

1. **Program: Guess my number**
   Class name: Guess my number
   Design and implement a guessing game program. The program picks a random number between 1 and 10. The program uses a 'while' loop that forces the user to keep guessing a number until they guess the mystery number correctly.
   CHALLENGE: Tell the user if their guess was too high or too low before they guess again.

   Example:
   I've chosen a number. Begin guessing.

   Enter your guess: 5
   Wrong, guess again: 3
   Wrong, guess again: 8

   You guessed it! The mystery number was 8.

2. **Program: Secret pass code**
   Class name: SecretPassCode
   Design and implement a program that is password protected. When the user runs the program, they are prompted to enter the secret pass code. The pass code is a 7 character string that you choose. The user is not allowed to move past this step until they enter the correct pass code. When the user enters an incorrect password, they receive a message that states "Invalid Pass Code" and are allowed to continue trying a new pass code until they correctly guess the pass code. If they enter the correct pass code, they receive a message that says "Valid Pass Code. Access Granted."

   CHALLENGE: Allow the user only three chances to enter the secret pass code. After the third failed attempt, the program states "Maximum attempts reached."

   Example:
   Enter the secret pass code: HistoryRocks
   Invalid Pass Code.
   Enter the secret pass code: EnglishRocks
   Invalid Pass Code.
   Enter the secret pass code: CSRocks
   Valid Pass Code. Access Granted.

3. **Program: Checkout Counter**
   Class name: CheckoutCounter
   Design and implement a program that simulates a checkout counter. The cashier will be greeted by the program and then asked for the cost of the first item. The cashier enters the cost of the first item. The cashier is then asked to enter the cost of the second item, and so on. When the cashier has no more items to enter, they enter 0 (zero) as the cost of the item. The program then displays how many items where checked out and the total cost of all of the items.

   Example:
   Welcome to Jiffy Fast Food Market.
   Enter the cost of the first item: 2.99
   Enter the cost of another item: 3.79
   Enter the cost of another item: 15.29
   Enter the cost of another item: 0

   The customer bought 3 items. The total is $22.07.

4. **Program: Darts 301**
   Class name: Darts301
   Design and implement a program that simulates the dart game 301. The user starts with 301 points. As long as the points are greater than or equal to 0, a dart is thrown. For this program, if the points reach a negative number, then the game is over. (In the real game of 301, you have to reach exactly zero.) A valid dart for this game is a random number between zero and 50. Display the current dart value and the running score after each throw.

   Example:
   Welcome to the Dart game 301!
   Score = 301

   Dart = 20, Score = 281
   Dart = 6, Score = 275
   Dart = 25, Score = 250
   :
   :
   Dart = 5, Score = 11
   Dart = 15, Score = -4
   Game Over

Computer Science 2 **Name** _____
Unit 7 The 'while' loop **Quiz #1 While loop analysis**

Predict the output of this program.

```
public class Mystery
{
  public static void main(String[] args)
  {
    int number = 0;
    while (number < 5)
    {
      System.out.println(number);
      number = number + 1;
    }
  }
}
```

Memory Storage

Predict the output of this program.

```
public class Mystery
{
  public static void main(String[] args)
  {
    int x = 10;
    while (x > 6)
    {
      System.out.println(Math.pow(x,2));
      x = x - 1;
    }
  }
}
```

Memory Storage

Computer Science 2
Unit 7 The 'while' loop

Name _____
Quiz #2 While loop analysis

Predict the output of this program.

```
public class Mystery
{
  public static void main(String[] args)
  {
    int y = 2;
    while (y < 7)
    {
      System.out.println(y);
      y = y + 1;
    }
  }
}
```

Memory Storage

Predict the output of this program.

```
public class Mystery
{
  public static void main(String[] args)
  {
    int number = 7;
    while (number >= 3)
    {
      System.out.println(Math.pow(number,2));
      number = number - 1;
    }
  }
}
```

Memory Storage

Computer Science 2
Console Programming
Unit 8: Arrays

Things to learn while writing these programs:

An array is variable that can store a list of values of the same type. When you want to write a program and you have a list of 'the same type of thing', you may consider using an array
The size of the array must be declared before using the array
An array cannot be 'resized'. It cannot be made longer or shorter once it is made.
The first location of an array is index zero, the second location is index one, the third location is index two, and so on.
An array can be made of any type (int, double, String, boolean, etc.) Examples int[] scores = new int[6] // creates an array called scores that can store up to 6 int values // the locations are: scores[0], scores[1], scores[2], scores[3], scores[4], and scores[5] double[] GPAs = new double[8] // creates an array called GPAs that can store up to 8 doubles String[] names = new String[15] // an array called names that can store up to 15 String values
The length of an array is its total number of locations. It can be found using .length
Uninitialized locations in an array automatically receive the default value for that type Default int values are 0. Default double values are 0.0. Default boolean values are false and default String values are null
To 'iterate' or 'traverse' through an array means to process each location in the array
The storage locations in an array are filled with 'elements'
To 'fill' an array means to assign values to the locations
An element of an array is a value that is stored in an array
'Hard-coding' an array means that the programmer fills the array with values (not the user)

Required Programs (You must do each of these!)
1. Here are my friends
2. Roster
3. Top Songs
4. Movie Ratings
5. Extra program of your own design

1. **Program: Here are my friends**
   Class name: HereAreMyFriends

Design and implement a program that displays the names of five friends. Create an array of Strings that can hold 5 elements. Hard-code the names of your friends into the array. After each location has a name, display each name using the array.

Example:
Here are my friends:
Superman
Batman
Batwoman
Robin
The Joker

2. **Program: Roster**
   Class name: Roster

Design and implement a program that displays the roster for a team. You can pick the team (any sport). Create an array that is the appropriate size for your team. Hard-code the names (first and last names) of the team, then display your team.

Example:
Here is my roster:
Aaron Rodgers
Jermichael Findley
James Starks
John Kuhn
:
:

3. **Program: Top Songs**
   Class name: TopSongs

Design and implement a program that asks the user to enter their top 7 songs titles. The program stores each song title into an array after it has been entered. When the user is done entering all 7 songs, the program displays the list.

Example:
Enter a song title #1: Back in Black
Enter a song title #2: Who are you?
Enter a song title #3: Ain't that a shame
Enter a song title #4: Right Now!
Enter a song title #5: Baba O'Reilly

Enter a song title #6: I want to hold your hand
Enter a song title #7: Sweet Home Alabama

Here are your Top 7 Song Titles!
Back in Black
Who are you?
Ain't that a shame
Right Now!
Baba O'Reilly
I want to hold your hand
Sweet Home Alabama

4. **Program: Movie Ratings**
   Class name: MovieRatings

Design and implement a program that asks for ratings for a movie. The program should ask 5 people for their rating of a movie (1 = LOW, 10 = HIGH) and keep track of each of the ratings. After the five people have entered their ratings, the program should display a list of all of the ratings and then display the average.

Example:
Please rate the movie The Social Network: 8
Please rate the movie The Social Network: 10
Please rate the movie The Social Network: 7
Please rate the movie The Social Network: 8
Please rate the movie The Social Network: 9

Here are the ratings for The Social Network:
8 10 7 8 9

The average rating is 8.4.

Computer Science 2
Unit 8 Arrays

Name _____
Quiz #1 Arrays

Predict the output of this program.

```
public class Mystery
{
  public static void main(String[] args)
  {
      String[] myList = new String[4];
      myList[0]= "He";
      myList[1]= "She";
      myList[2]= "They";
      myList[3]= "Them";

      System.out.println(myList[1]);
      System.out.println(myList[3]);
      System.out.println(myList[2]);
      System.out.println(myList[0]);
  }
}
```

Memory Storage

_____

_____

_____

_____

_____

Predict the output of this program.

```
public class Mystery
{
  public static void main(String[] args)
  {
      int[] myArray = new int[5];
      for(int i = 0; i < myArray.length; i = i + 1)
         myArray[i] = 3*i;
      for(int j = 0; j < myArray.length; j = j + 1)
         System.out.println(myArray[j]);
  }
}
```

Memory Storage

_____

_____

_____

_____

_____

Computer Science 2
Unit 8 Arrays

Name _____
Quiz #2 Arrays

Predict the output of this program.

```
public class Mystery
{
  public static void main(String[] args)
  {
       String[] myArray = new String[4];
       myArray[0]= "John";
       myArray[1]= "Paul";
       myArray[2]= "George";
       myArray[3]= "Ringo";

       System.out.println(myArray[2]);
       System.out.println(myArray[0]);
       System.out.println(myArray[3]);
       System.out.println(myArray[1]);
  }
}
```

Memory Storage

_____

_____

_____

_____

_____

Predict the output of this program.

```
public class Mystery
{
  public static void main(String[] args)
  {
     int[] listOfStuff = new int[4];
     for(int i = 0; i < listOfStuff.length; i = i + 1)
       listOfStuff[i] = 10-i;
     for(int j = 0; j < listOfStuff.length; j = j + 1)
       System.out.println(listOfStuff[j]);
  }
}
```

Memory Storage

_____

_____

_____

_____

_____

Computer Science 2
GUI Programming
Unit 1: Basic input and output using Dialog Windows

Things to learn while doing this assignment

GUI stands for Graphical User Interface
GUI is more fun for the user than the console
GUI allows for programs to be 'event driven' using the mouse (the console does not)
Dialog windows are part of the JOptionPane Class. Here is a tutorial: (http://docs.oracle.com/javase/tutorial/uiswing/components/dialog.html)
The required import statement is: `import javax.swing.JOptionPane;`
To display a message with an OK button: `JOptionPane.showMessageDialog(null, "Computer Science Rocks!");`
To ask a question that requires a 'word' response: `String name = JOptionPane.showInputDialog("What is your name?");`

To ask a question that requires an 'integer' response:
`String stringValue = JOptionPane.showInputDialog("What is your age?");`
`// Now, convert the input to an integer`
`int age = Integer.parseInt(stringValue);`

To ask a question that requires a 'double' response:
`String stringValue = JOptionPane.showInputDialog("What is your gpa?");`
`// Now, convert the input to a double`
`double gpa = Double.parseDouble(stringValue);`

To display a question with a 'YES/NO' option:
`int response = JOptionPane.showConfirmDialog(`
`  null,`
`  "This is where the question goes",`
`  "This is the title of the window",`
`  JOptionPane.YES_NO_OPTION);`

If the user clicks YES, then response has a value of 0
If the user clicks NO, then response has a value of 1
You MUST close the interactions pane using a `System.exit(0);`

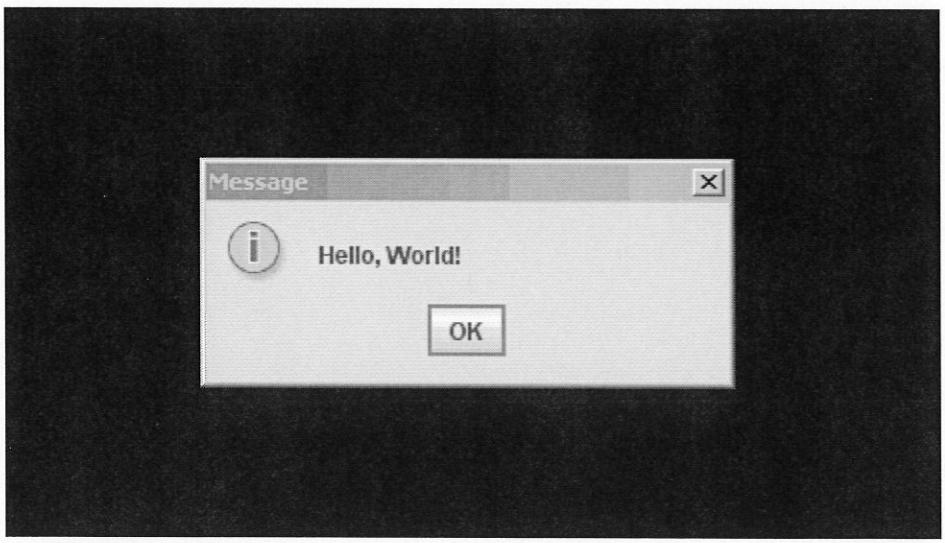
Required Programs (You must do each of these!)

1. Hello World GUI
2. Entering a String GUI
3. Entering an integer GUI
4. Entering a double GUI
5. Yes or No GUI
6. Extra program of your own design

1. Program: Hello World GUI
Class name: HelloWorldGUI

Copy and paste the following program into a new file in DrJava. Compile and run it.
Requirement: Modify it so that it says: "I just made my first GUI program!"

```
import javax.swing.JOptionPane;

public class HelloWorldGUI
{
  public static void main(String[] args)
  {
     JOptionPane.showMessageDialog(null, "Hello,World!");

     System.exit(0);
  }
}
```



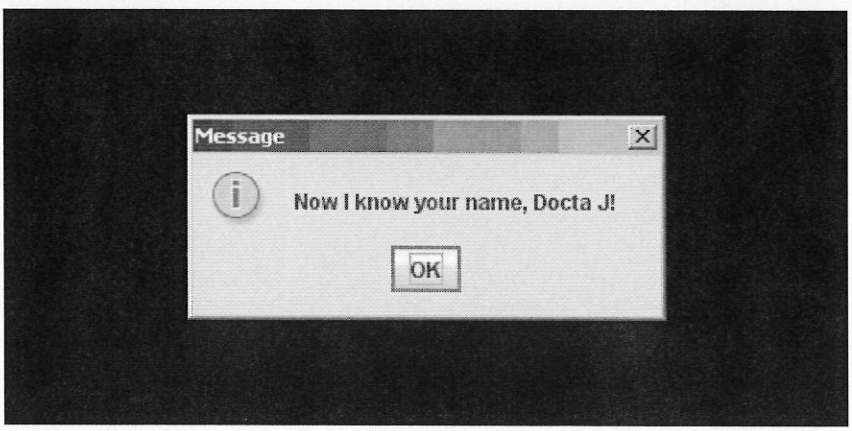
2. Program: Entering a String GUI

Class name: EnteringAStringGUI

Copy and paste the following program into a new file in DrJava. Compile and run it

Requirement: Modify the program so that it asks for the user's last name in a separate dialog box.

```
import javax.swing.JOptionPane;

public class EnteringAStringGUI
{
  public static void main(String[] args)
  {
   String name = JOptionPane.showInputDialog("What is your name?");
   JOptionPane.showMessageDialog(null, "Now I know your name, " +
                                 name + "!");

   System.exit(0);
  }
}
```



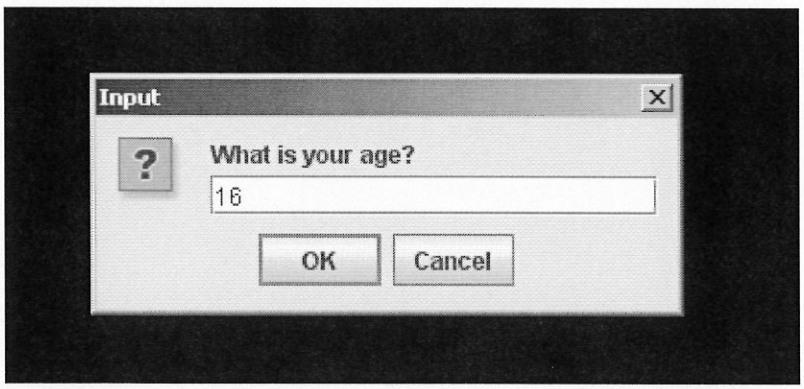
3. Program: Entering an integer GUI

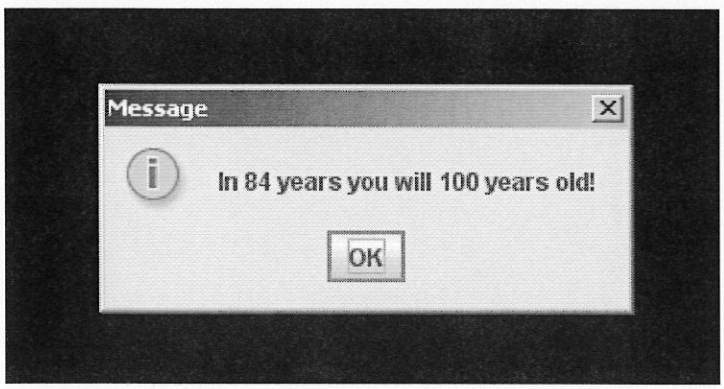
Class name: EnteringAnIntGUI

Copy and paste the following program into a new file in DrJava. Compile and run it

Requirement: Modify the program so that it asks a different question.

```
import javax.swing.JOptionPane;

public class EnteringAnIntGUI
{
  public static void main(String[] args)
  {
   String stringValue = JOptionPane.showInputDialog("What is your
age?");
   int age = Integer.parseInt(stringValue);  // convert the input to
an integer

   JOptionPane.showMessageDialog(null, "In " + (100 - age) + " years
you will 100 years old!");

   System.exit(0);
  }
}
```





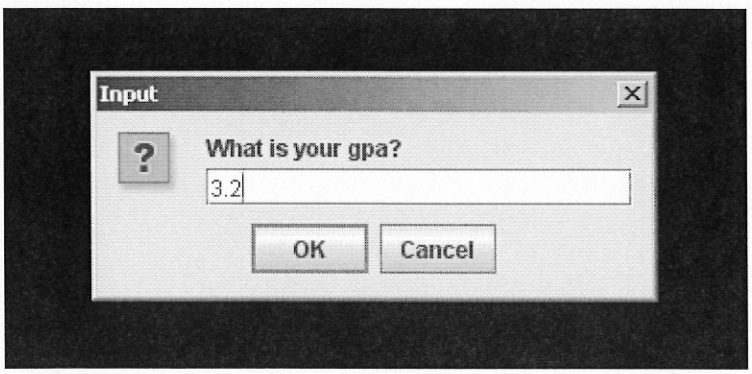
4. Program: Entering a double GUI

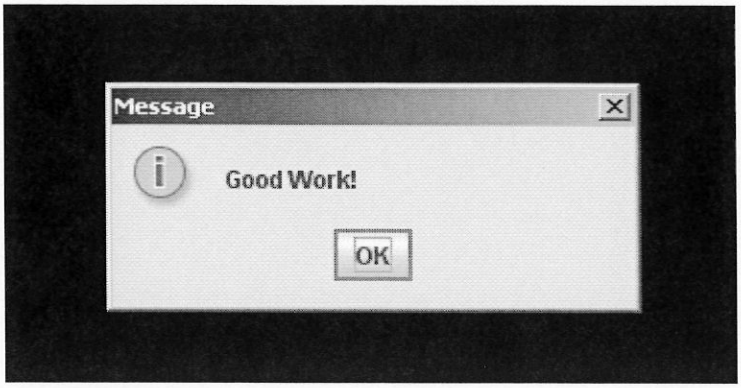
Class name: EnteringAnDoubleGUI

Copy and paste the following program into a new file in DrJava. Compile and run it

Requirement: Modify the program so that it asks a different question.

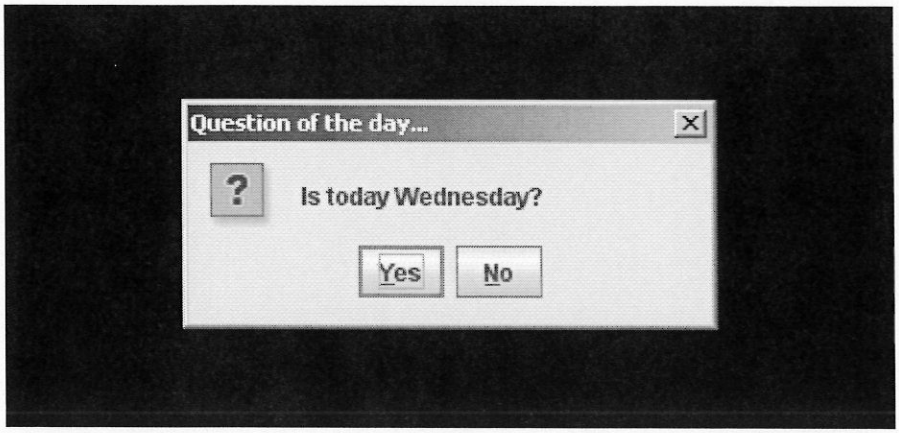
```
import javax.swing.JOptionPane;

public class EnteringADoubleGUI
{
  public static void main(String[] args)
  {
    String stringValue = JOptionPane.showInputDialog("What is your
gpa?");
    double gpa = Double.parseDouble(stringValue);  // convert the
input to a double

    if (gpa >= 3.0)
      JOptionPane.showMessageDialog(null, "Good Work!");
    else
      if (gpa >= 2.0)
      JOptionPane.showMessageDialog(null, "Alright, that's ok.");
    else
      JOptionPane.showMessageDialog(null, "Time to get to work!");

    System.exit(0);
  }
}
```

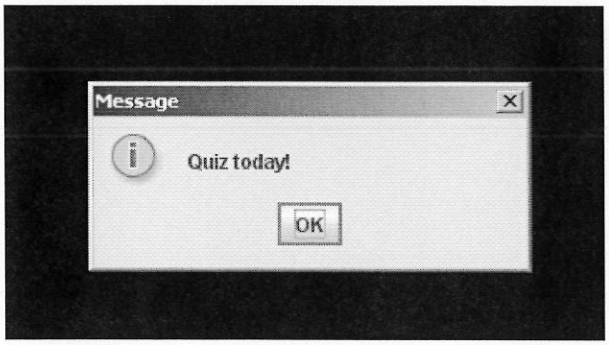


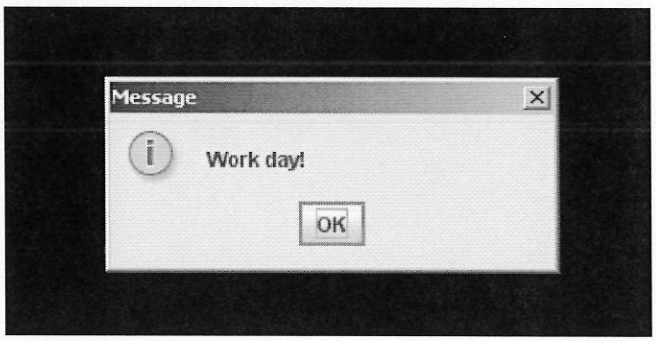


5. Program: Yes or No GUI
Class name: YesOrNoExample
Copy and paste the following program into a new file in DrJava. Compile and run it
Requirement: Modify the program so that it asks a different question.

```
import javax.swing.JOptionPane;

public class YesNoGUI
{
  public static void main(String[] args)
  {
    int response = JOptionPane.showConfirmDialog(
         null,"Is today Wednesday?","Question of the day...",
         JOptionPane.YES_NO_OPTION);

    if (response == 0)
      JOptionPane.showMessageDialog(null, "Quiz today!");
    else
      JOptionPane.showMessageDialog(null, "Work day!");

    System.exit(0);
  }
}
```


Question of the day...
Is today Wednesday?
Yes
No


Message
Quiz today!
OK


Message
Work day!
OK

Computer Science 2
GUI Unit Quiz

Name _____
Quiz #1 GUI

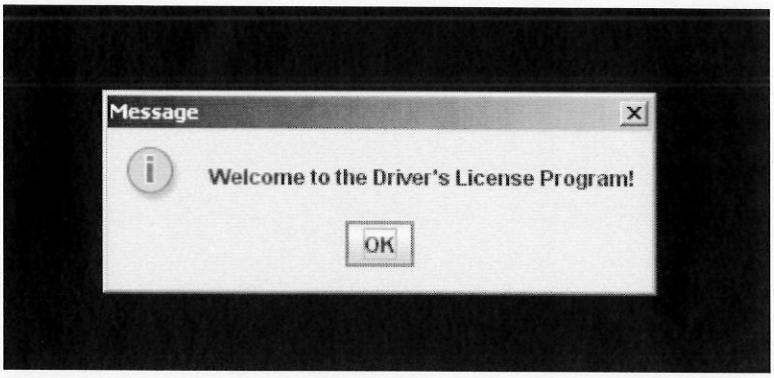
Write a GUI program that determines whether or not a person can apply for their driver's license.

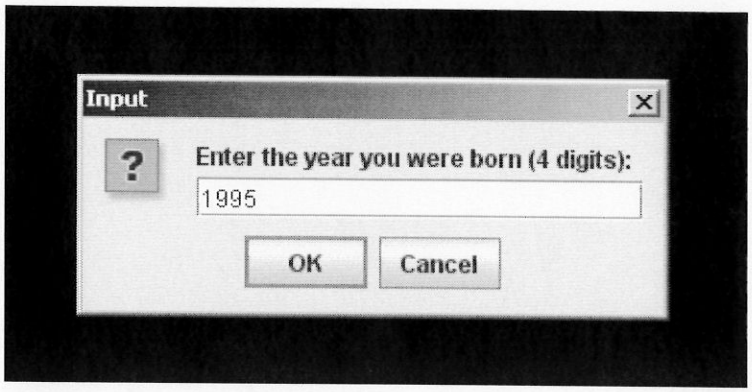
The user must enter:

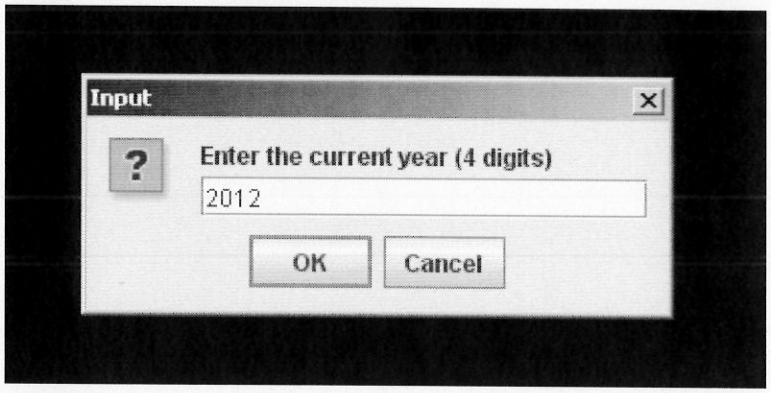
- Their name
- The year they were born
- The current year

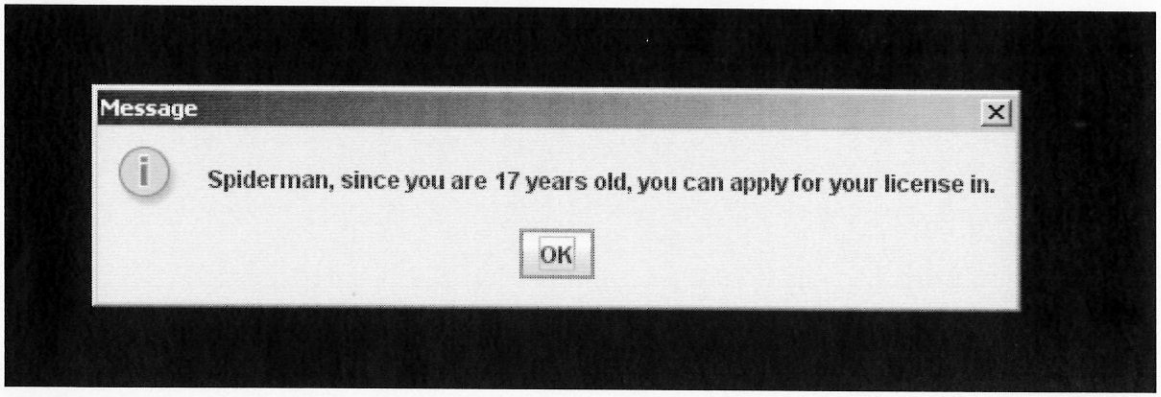
If they will turn 16 this year, then they can apply for their license. Otherwise, they cannot.

Follow the example screen shots to write your program.

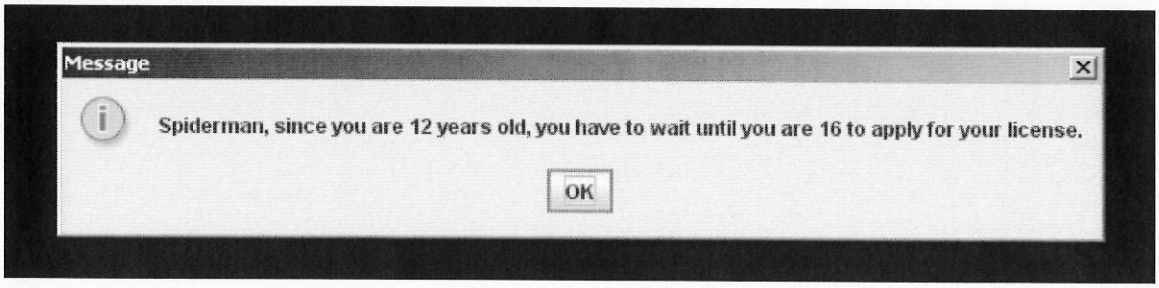








If Spiderman is not old enough, then this is the result:



Computer Science 2
GUI Unit Quiz

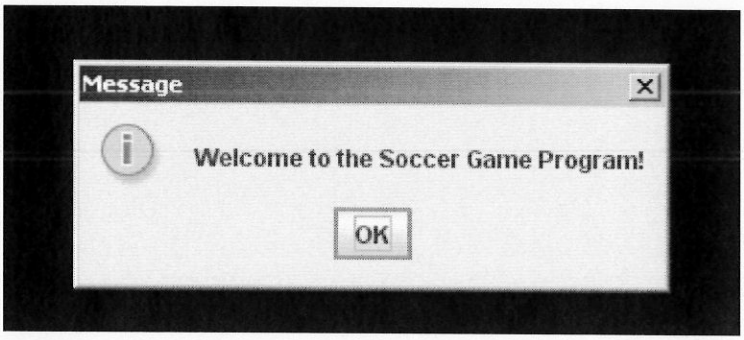
Name _____
Quiz #2 Soccer Game

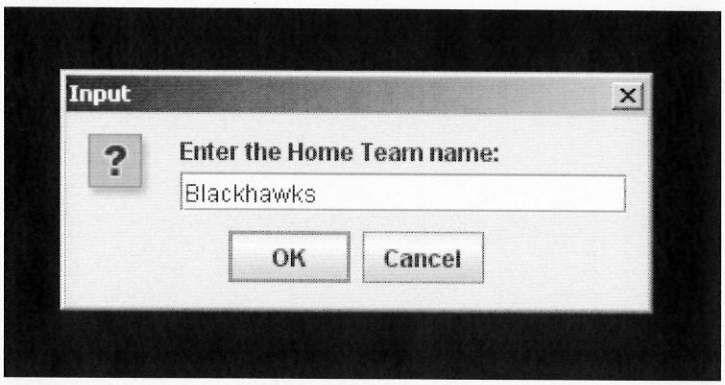
Write a GUI program that determines the winner of a soccer game.

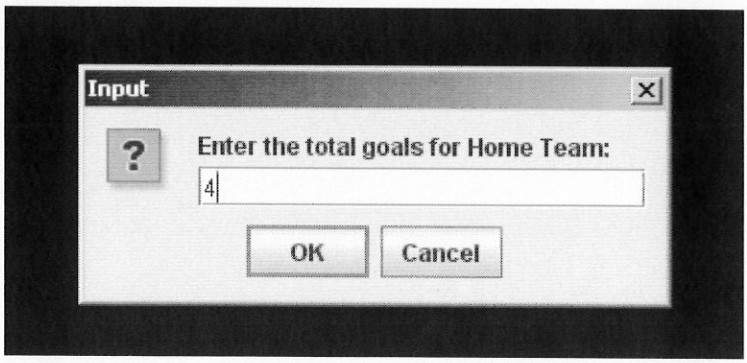
The program must ask the user for:

- The name of the Home Team
- How many goals the Home Team made
- The name of the Visitor Team
- How many goals the Visitor Team made

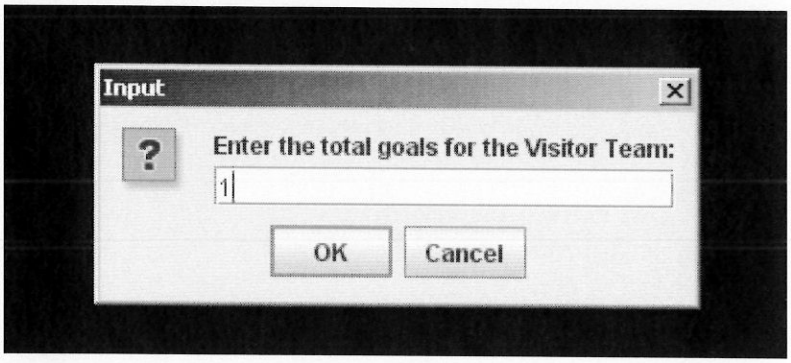
The program must tell the user which team won or if the game ended in a tie.

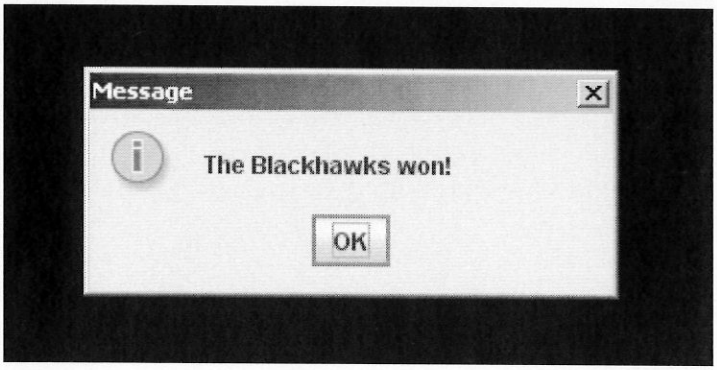







Input
Enter the Visitor Team name:
Eagles
OK
Cancel


Input
Enter the total goals for the Visitor Team:
1
OK
Cancel


Message
The Blackhawks won!
OK

Computer Science 2
GUI Programming
Unit 2: Converting from Console to GUI

Required Programs (You must do each of these!)
1. Choose any console program and convert it from Console to GUI
2. Choose any console program and convert it from Console to GUI
3. Choose any console program and convert it from Console to GUI

Computer Science 2
GUI Programming
Unit 3: Menu-Driven Programs

Things to learn while doing this assignment

Menu-Driven programs provide the user with choices. These choices must be declared in an array.

A Main Menu is required for menu-driven programs.

This is an example of the Main Menu Display

```
// This statement declares an array and hard-codes the menu choices
Object[] possibilities = {"Thunderstruck", "Sweet Home Alabama", "VanHalen-RightNow","Exit"};
// This statement stores the users choice in the String variable 'choice'
 String choice = (String)JOptionPane.showInputDialog(
                                        null,  // use null if frame is not known
                                        "Choose the song you want to play",  // Menu title
                                        "iTunes Player",  // Window title
                                        JOptionPane.PLAIN_MESSAGE,  // Icon
                                        null,  //
                                        possibilities,  // Array of choices in the menu
                                       "Thunderstruck");  // The option that is highlighted (usually the 1st one)
```

To perform the task that was selected by the user, you much compare choice to menu options.

Example

```
 if (choice.equals("Thunderstruck"))
     {
      String songName = "Thunderstruck.wav";
      Sound.play(songName);
      JOptionPane.showMessageDialog(null, "Playing " + "Thunderstruck", "CS2 and company",
                          JOptionPane.INFORMATION_MESSAGE, null);
     }
```

The basic **algorithm for the main menu** follows the input algorithm for a while loop:

```
Step 1:  Present the main menu to the user
Step 2:   While (their choice is not equal to exit)
          {
             If (their choice is the first option)
                Perform first option
             If (their choice is the second option)
                Perform second option
             … and so on for each option
Step 3:       Present the main menu to the user.  // This is very important!!!!!!!!!
          }
```

Required Programs (You must do each of these!)
1. Simple Calculator
2. Music Player
3. Student Creation

1. **Program: Simple Calculator**
   Class name: SimpleCalculator

   Copy and paste the Simple Calculator folder to your h: drive. Compile and Run the SimpleCalculator program. Fix the program by completing all sections.

   Be sure to
   - Modify all variables so that they are appropriate
   - Modify all comments so that they are appropriate
   - Use appropriate icons for each operation (these images MUST be stored in the Simple Calculator folder.)

2. **Program: Music Player**
   Class name: MenuDrivenMusicPlayer

   Copy and paste the Music Player folder to your h: drive. Compile and Run the MusicPlayer program. Modify the program by replacing my songs with 3 or more of your own songs.

   Be sure to
   - Use .wav music files
   - Store the music files in the Music Player folder

3. **Program: Student Creation**
   Class name: StudentCreation

   - Modify one of the examples to create your own Menu-Driven program.
   - You must have 3 or more options
   - Be Creative!

Computer Science 2 **Name** _____
Game Project 1 - The SIMPLE DEMO Project

MODIFICATION	POINT VALUE	Points Earned
1. Make the robot move left (without transporting)	1	
2. Make the robot move up (without transporting)	1	
3. Make the robot move down (without transporting)	1	
4. Make the robot move right (with transporting)	1	
5. Make the robot move left (with transporting)	1	
6. Make the robot move up (with transporting)	1	
7. Make the robot move down (without transporting)	1	
8. Make the robot face left when moving left	1	
9. Make the robot face right when moving right	1	
10. Make the robot face up when moving up	1	
11. Make the robot face down when moving down	1	
12. Green square is changed to a different color	1	
13. Gridlines are a different color other than black	1	
14. Title of window is changed to "Game Project 1"	1	
15. Make center square impenetrable from left	1	
16. Make center square impenetrable from right	1	
17. Make center square impenetrable from below	1	
18. Make center square impenetrable from above	1	
19. Change the dimensions of the grid	1	
20. Change the 'hero' to a different image (Mario)	1	
TOTAL POINTS FOR 'SIMPLE DEMO' PROJECT	**20**	
CHALLENGES:		
Create a score and put it in the window title	1-3	
Update the score when you move through certain squares	1-3	
Add music so it plays while you play	1-3	
Count the number of moves made and display it in the title	1-3	
Anything else you can think of	1-3	

Computer Science 2 **Name** _____
Unit 10 - The TETRIS Project

MODIFICATION	POINT VALUE	Points Earned
Change the color for shape 0	1	
Change the color for shape 1	1	
Change the color for shape 2	1	
Change the background color	1	
Change the gridlines to another color	1	
Change the title of the main dialog box	1	
Create a new shape 3	3	
Create a new shape 4	3	
Create a new shape 5	3	
TOTAL POINT FOR PROJECT	**15**	
CHALLENGES:		
Keep score and update window title	1-3	
Move a shape to the bottom immediately after press 'space' bar	1-3	
Show what shape is coming next	1-3	
Tell the player when they lose	1-3	
Tell they player when they won	1-3	
Add music	1-3	
Think of something else	1-3	

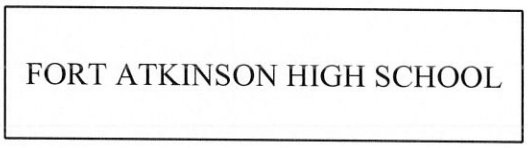
Computer Science 2
Quarter 3 Exam

Name _____

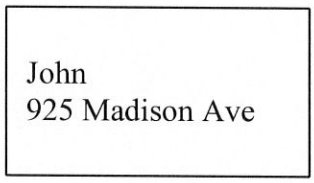
Complete as many of the following programs as you can in the class period. Use standard java style conventions for variable names and types. You may receive partial credit for programs that are no finished.

PRINT OUT EACH – PUT YOUR NAME IN A COMMENT AT THE TOP

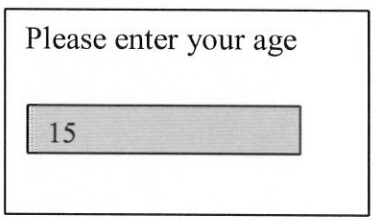
Program 1 - Sign (5 points)
Write a program that will print the words FORT ATKINSON HIGH SCHOOL in a GUI textbox on the screen.

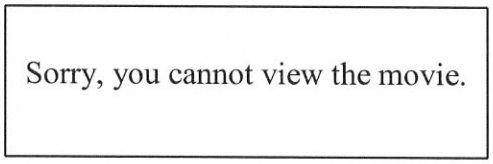


Program 2 – Label (10 points)
Write a program that will ask the user for their *first name*, *street address*, and then print their responses in a GUI textbox on two different lines.



Program 3 – Movie (15 points)
Write a program that decides if the user can attend an R rated movie. Only people with ages 17 and over can view the movie. Ask the user for their *age* and then tell them if they are allowed to view the movie.





Program 4 – All About 3 (15 points) *Non GUI
Write a program that displays the following information about the number 3.

The square root of 3 rounded to 3 decimal places is _____.
The sine of 3 degrees rounded to 4 decimal places is _____.
The number 3 raised to the 10^{th} power is _____.
The arc-cosine of the number 3 rounded to 1 decimal place is _____.

Program 5 – Grader (20 points) *Non GUI
Write a program that decides the grade for a student with three scores. Here are the three scores: score 1 = 87, score 2 = 84, and score is 68. Write a *console*-based program that averages the three scores and then displays the grade for the student. Use the following grading scale to determine the student's grade. Remember that this is NOT a GUI program. Use *System.out.println()* statements to display the student's average and their grade.

A	At least 90 - 100
B	At least 80 – less than 90
C	At least 70 – less than 80
D	At least 60 – less than 70
F	Less than 60

Computer Science 2 **Name** _____ **Hr** ____
Semester Exam – Polygon App

Write each of the programs and save them to your h: drive in a folder called 'Semester Exam.' You will be graded on the total number of points that you accumulate, and will be given partial credit for programs that are not completed, yet show progress.

Write an application to calculate the area of a regular polygon.
The formula is A = ½ a p, where a is the apothem and p is the perimeter. The lengths of a side and the apothem *could* be decimals, but the number of sides must be a whole number.

Sample input and output:

```
Welcome to the Polygon Area App.

Enter the number of sides for the polygon:  5
Enter the length of one side of the polygon:  8
Enter the apothem of the polygon:  5.5

The area of this polygon is 40.
```

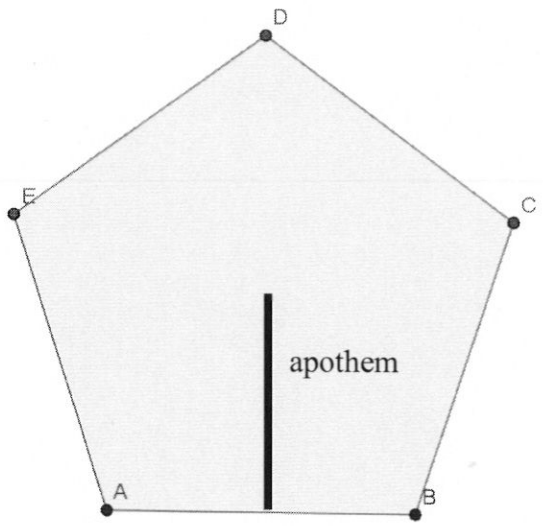
Program 1:
Console (20 points)
Write the program in DrJava and use a Scanner for input and System.out.println() for output.

Program 2:
GUI (20 points)
Write a program in DrJava that uses JOptionPane dialog boxes to input and output the values.

Program 3:
NetBeans GUI Builder (20 points)
Write the program using Netbeans. Create a new project that allows the user to enter the information in textboxes and then press a button that calculates and displays the answer the answer in a textbox. The program should have proper labels and appropriate variable names.

Example Data to test your class:
Apothem = 5.5
Side Length = 8
Number of Sides = 5

Area = 110
Perimeter = 40



Computer Science 2 **Name** _____ **Hr** ____
Semester Exam – Height of a ball dropped from a tower

Write each of the programs and save them to your h: drive in a folder called 'Semester Exam.' You will be graded on the total number of points that you accumulate, and will be given partial credit for programs that are not completed, yet show progress.

Write an application to calculate the height of a ball dropped from a tower.
When a ball is dropped from the top of a tower, it's height at any time can be calculated using the formula:

$$H = -16*T^2 + B$$

where T is the number of seconds after the ball is dropped and B is the height of the tower. For this problem, the tower height should be a whole number, where the number of seconds should be a decimal number.

Sample input and output:

```
Welcome to the Ball Height Predictor.

Enter the height of the tower (in feet):  300
Enter the time (in seconds):  2.7

The height of the ball would be 183.36
```

Program 1:
Console (20 points)
Write the program in DrJava and use a Scanner for input and System.out.println() for output.

Program 2:
GUI (20 points)
Write a program in DrJava that uses JOptionPane dialog boxes to input and output the values.

Program 3:
NetBeans GUI Builder (20 points)
Write the program using Netbeans. Create a new project that allows the user to enter the information in textboxes and then press a button that calculates and displays the answer the answer in a textbox. The program should have proper labels and appropriate variable names.