

AP Computer Science Mr Hanley

[The Hood](#)

Summer Assignment

Ver: 2.01

Last Updated: **8/9/2021 4:46 PM**

Assignment 0: Summer Assign

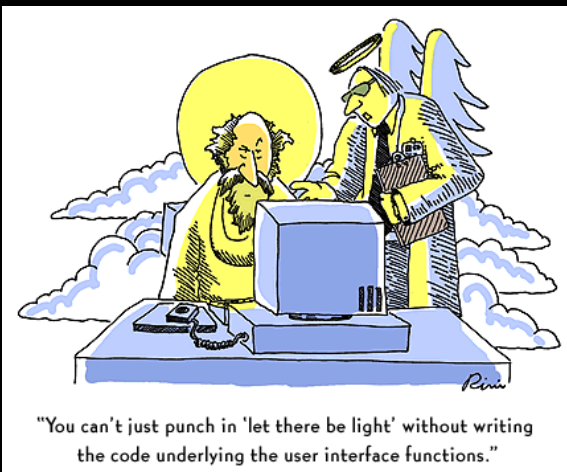
Binary



Ones Comp



Twos Comp



Welcome to the AP Computer Science Program!

This is a challenging and fun course as we delve deeper into computer science. Here you will find summer assignments to be completed throughout the summer...approximate dates are given in this document.

The two articles are from the Oracle page and they describe the basic concepts of classes and objects. Here are the URL's

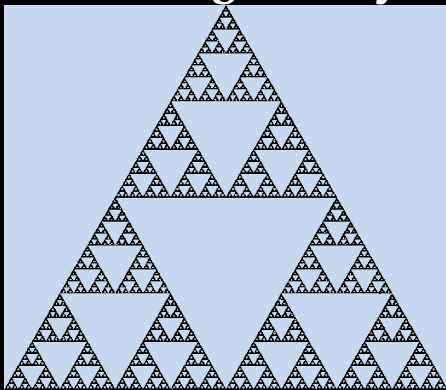
<https://docs.oracle.com/javase/tutorial/java/concepts/object.html>

<https://docs.oracle.com/javase/tutorial/java/concepts/class.html>

1. Object Oriented Presentation

Prepare a 6-8 slide powerpoint presentation or google slides on the articles describing the differences. If necessary, use the internet and computer books to explore the differences between classes and objects, etc. Give an example of a class and an example of an object.

2. Triangle Analyzer



Write a program that takes as inputs the lengths of three sides of a triangle and displays; first, whether or not the 3 values represent a triangle and then whether the triangle is scalene, isosceles, or equilateral.

Useful facts:

In a triangle, the longest side must be less than the sum of the other two sides.

A scalene triangle has all sides unequal.

An isosceles triangle has two sides equal.

An equilateral triangle has all sides equal.

(Pictured above is Sierpinski's Triangle, for fun try to make it using Processing this summer!!!!)

HINTS for Testing: When testing, make sure you test with all sides the same and your program doesn't report the triangle to be both isosceles and equilateral.

3. Craps!!!!!!!!!!!!!!



DO NOT USE Math.random() for this assignment!!!!

This assignment needs to have a random number generator using the java.util.Random class

I will be using a [seed](#) to test this.

Only generate numbers when you are using them to roll dice.

See below for the results expected given a seed of 10!

In the game of craps, a player rolls 2 six sided dice and adds up the sum. The first roll is special; a player can either win or lose instantly on the first roll.

DO NOT USE `nextInt(11) + 2`; **THIS DOES NOT SIMULATE PROBABILITY AND THE LIKELYHOOD OF 7 COMING UP MOST OFTEN!!!**

If the first roll is a 7 or 11, **INSTANT WINNER**. The round is over. If a player rolls a 2, 3 or 12, **INSTANT LOSER**. If none of the above are rolled, the player goes into a phase where they continually roll 2 dice until one of two things happens;

- 1) the player rolls the initial number (known as the point) the player **WINS**
- 2) the player rolls a 7, the player **LOSES**.

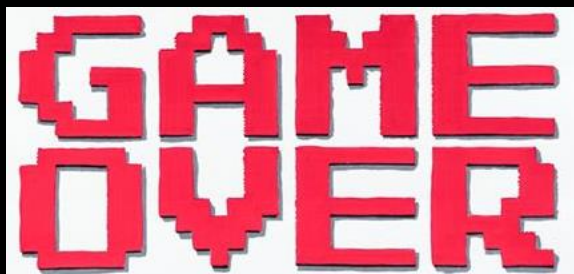
1. Examples of dice roll sequences and results

- i. Roll a 7 on first roll -> WIN
- ii. Roll an 11 on first roll -> WIN
- iii. Roll a 2 on first roll -> LOSE
- iv. Roll a 3 on first roll -> LOSE
- v. Roll a 12 on first roll -> LOSE
- vi. Roll a 5, 9, 12, 6 and then a 5 to win since you rerolled the point
- vii. Roll a 5, 9, 12, 6 and then a 7 to lose since you rolled a 7 when attempting to reroll the point
- viii. Roll 8, 2, 3, 11, 12, 9, 3, 3, 5, 9, 10, 8 to win since you rerolled the point

Make a craps game in the console or Swing that starts the player out with 100 chips. (or cookies or paper clips, bird eggs, whatever)

Before each game, the user is asked to make a wager. (by default, just wager 10%) Then allow the user to play the game by rolling the dice. Use the space bar or r key or something to allow them to see each roll. After the game is won or lost, subtract or add chips and then display

them to the screen. If the user is out of chips, then



Sample Results for Seed 10 from Alex Herman (2017) Project run:

You have 100 chips. How many do you want to bet?

1

Type 'r' to roll the dice.

r

You rolled a 5

Type 'r' to roll the dice again.

r

You rolled a 5

YOU WON!!!! You got 1 chip.

\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$ WINNER \$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$

You have 101 chips. How many do you want to bet?

2

Type 'r' to roll the dice.

r

You rolled a 10

Type 'r' to roll the dice again.

r

You rolled a 7

You crapped out and lost 2 chips.

=====LOSER=====

You have 99 chips. How many do you want to bet?

2

Type 'r' to roll the dice.

r

You rolled a 9

Type 'r' to roll the dice again.

r

You rolled an 8

Type 'r' to roll the dice again.

r

You rolled a 5

Type 'r' to roll the dice again.

r

You rolled a 5
Type 'r' to roll the dice again.
r
You rolled a 7
You crapped out and lost 2 chips.
=====LOSER=====

You have 97 chips. How many do you want to bet?
3
Type 'r' to roll the dice.
r
You rolled a 7
YOU WON!!!! You got 3 chips.
\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$ WINNER \$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$

You have 100 chips. How many do you want to bet?
2
Type 'r' to roll the dice.
r
You rolled an 8
Type 'r' to roll the dice again.
r
You rolled a 7
You crapped out and lost 2 chips.
=====LOSER=====

You have 98 chips. How many do you want to bet?
2
Type 'r' to roll the dice.
r
You rolled an 8
Type 'r' to roll the dice again.
r
You rolled a 6
Type 'r' to roll the dice again.
r
You rolled a 6
Type 'r' to roll the dice again.
r
You rolled a 6
Type 'r' to roll the dice again.
r
You rolled a 4
Type 'r' to roll the dice again.
r
You rolled a 4

Type 'r' to roll the dice again.
r
You rolled a 10
Type 'r' to roll the dice again.
r
You rolled a 6
Type 'r' to roll the dice again.
r
You rolled an 8
YOU WON!!!! You got 2 chips.
\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$ WINNER \$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$
You have 100 chips. How many do you want to bet?
10
Type 'r' to roll the dice.
r
You rolled a 4
Type 'r' to roll the dice again.
r
You rolled a 11
Type 'r' to roll the dice again.
r
You rolled a 2
Type 'r' to roll the dice again.
r
You rolled an 8
Type 'r' to roll the dice again.
r
You rolled a 12
Type 'r' to roll the dice again.
r
You rolled a 10
Type 'r' to roll the dice again.
r
You rolled a 12
Type 'r' to roll the dice again.
r
You rolled a 4
YOU WON!!!! You got 10 chips.
\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$ WINNER \$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$
You have 110 chips. How many do you want to bet?
3
Type 'r' to roll the dice.
r

```
You rolled a 4
Type 'r' to roll the dice again.
r
You rolled a 4
YOU WON!!!! You got 3 chips.
$$$$$$$$$$$$$$$$ WINNER $$$$$$$$$$$$$$$$$
```

HINTS for Testing: If you seed the Random number generator object, then you will get the same sequence of pseudo random numbers each run of the program. For example, `Random r = new Random(100);` will cause all calls to `r.nextInt(n);` to give the same sequence. This helps you find the repeated case you are looking for without testing forever. Be sure you've covered all cases.

NOTE: Console programmers will be writing a loop while Swing users will have to keep track of whether it is a reroll or initial roll. Swing utilizes its own loop to keep checking for any events that might occur and report those events to the program handlers. Therefore, although there is a loop, it has been created by the authors of the Swing GUI developers. To deal with this, use a global variable inside your Frame to determine if the next roll is an initial roll or reroll.

For example,

```
public CrapsFrame extends javax.swing.JFrame{
    boolean firstRoll = true;

}
}
```

This variable can be used to inform the `actionPerformed` method inside the `CrapsFrame` when a roll is a fist roll and when it is a reroll. Just flip the boolean variable.

NOTE: Swing GUI's MUST BE ABLE to show the sequence of rolls for

ease of testing. To do this, use a JTextArea that can show 6 rows of numbers.

Each roll can be appended to the JTextArea so I can see the sequence of rolls.

4. Perfect Number Decider



A perfect number is a positive integer such that the sum of the divisors equals the number. Thus, $28 = 1 + 2 + 4 + 7 + 14$ is a perfect number. If the sum of the divisors is less than the number, it is deficient. If the sum exceeds the number, it is abundant. Write a program that takes a positive integer as input and displays a message box that indicates whether the number entered is perfect, deficient, or abundant. Your program should define the following three methods:

NOTE: MAKE SURE YOU HAVE ALL METHODS AND PARAMETERS, YOU WILL NOT RECEIVE FULL CREDIT UNLESS YOU STRUCTURE YOUR PROGRAM AS YOU SEE BELOW!!

```
main
boolean isDivisor (int num, int divisor)
int divisorSum (int number)
```

The method isDivisor returns true if the divisor parameter is a divisor of the number parameter, and false otherwise. The divisorSum method

uses `isDivisor` to accumulate and return the sum of the proper divisors of the number parameter. Be sure to design and test the program incrementally, that is, verify that `isDivisor` works correctly before using it in `divisorSum`.

From a console program, the three methods might look like this;

```
/* prompts the user for a number and then reports whether the
number is perfect, deficient or abundant , calls the divisorSum method
to find the number of proper divisors for any number*/
```

```
public static void main(String[] args) {
    //How do you call divisorSum from here?
    // int sum = divisorSum(num); //calls method and stores
return value in sum
}
```

```
/* takes in an integer and calls isDivisor in a loop to determine
whether or not each possible divisor evenly divides the number*/
```

```
public static int divisorSum (int number)
}
```

```
/* takes in an two integers, returns true if the second is a divisor of
the first*/
```

```
public static boolean isDivisor (int num, int divisor) {
}
```

```
public static void main()
```

Gather input from user
Call divisorSum to find out sum of proper divisors



```
public static int divisorSum(int num)
```

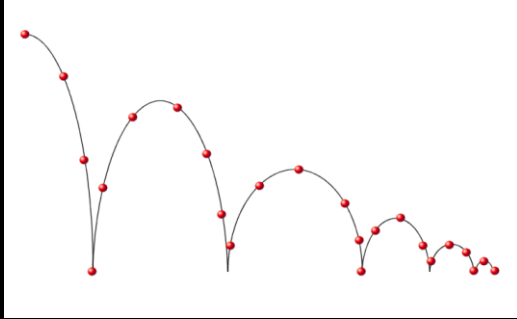
Loops through possible candidates, calls isDivisor to see if each potential divisor is actually a divisor
LOOP MULTIPLE TIMES TO CALL isDiv



```
public static boolean isDivisor(int num, int div)
```

returns true or false depending upon the relationship between num and div

5. Bounce Calculator



A standard experiment is to drop a ball to see how high it bounces. Once the "bounciness" of the ball is determined, the ratio gives a bounciness index. For example, if a ball dropped from a height of 10 feet bounces 6 feet high, the index is 0.6 and the total distance traveled by the

ball is 16 feet after **one bounce**. If the ball continues bouncing, the distance after two bounces would be $10 + 6 + 6 + 3.6 = 25.6$ feet.

Note that the distance traveled for each bounce is the distance to the floor plus 0.6 of that distance as the ball comes back up.

The ball, therefore, **hovers in the air**. Write a program that takes as inputs the initial height of the ball (in feet), the index of the ball's bounciness, and the number of times the ball is allowed to continue bouncing. The program should output the total distance traveled by the ball.

At some point in the process, the distance traveled by the ball after a bounce might become negligible, for example, less than 0.001 feet . If that stage is reached, terminate the process and output the total distance. (This could happen if for example, we attempted to drop a ball for 10,000 bounces).

Here are run-throughs that will be helpful to you. For Chris Du, Rahul Shah and Jason Canterbury (Fall 2017???)!!!!

Case #	Initial Height	Bounciness	Bounces	Distance	Breakdown
1	50	.4	5	115.472	see below
2	10	.8	3	43.92	see below
3	1	.2	8	1.4992	only 4 and a half bounces
Make sure threshold set to .001 for case 3!!!					

EXAMPLE 1

What Height to Start?

50

Index of Bounce?(.1->.9)

.4

How many bounces?

5

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

Down:50.0

Up: 20.0

Down:20.0

Up: 8.0

Down:8.0

Up: 3.2

Down:3.2

Up: 1.2800000000000002

Down:1.2800000000000002

Up: 0.5120000000000001

115.47200000000001

EXAMPLE 2

What Height to Start?

10

Index of Bounce?(.1->.9)

.8

How many bounces?

3
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

Down:10.0
Up: 8.0

Down:8.0
Up: 6.4

Down:6.4
Up: 5.1200000000000001

43.92
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

EXAMPLE 3

What Height to Start?

1

Index of Bounce?(.1->.9)

.2

How many bounces?

8
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

Down:1.0
Up: 0.2

Down:0.2
Up: 0.04000000000000001

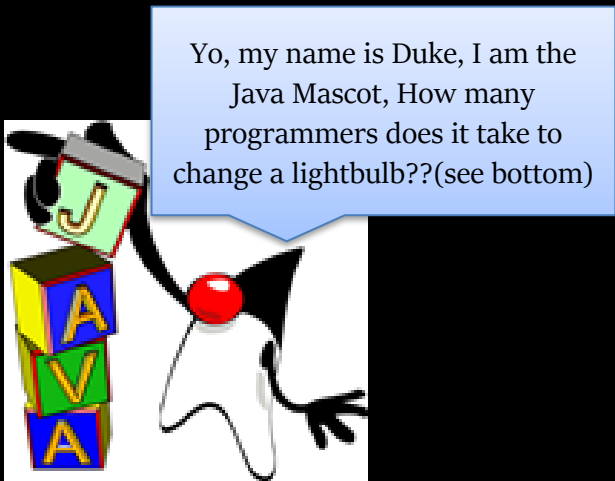
Down:0.04000000000000001
Up: 0.008000000000000002

Down:0.0080000000000000002
Up: 0.0016000000000000005

Down:0.0016000000000000005
Up: 3.20000000000000013E-4

Since up is less than 0.001 we outta here!
1.4992

@@



6. Create an account on the codingbat.com web site. This site allows you to type in the bodies of methods and see if they are correct online. Share your account out to mrhanleyc@hotmail.com under the prefs for codingbat. **Your first and last name in the preferences must be your actual first and last name so I don't have to guess who is who.**

CodingBat code practice
> CodingBat Home

Account Settings

Change password

Password must contain at least 6 characters

New Password

Teacher Share

Enter the email address of the teacher account. This will make your done page and solution code visible to that account.

Share To

Memo

Generally this is left blank. A teacher may ask you to fill this in.

Memo

Delete Account

Permanently delete account - cannot be undone







[privacy statement](#)

- a. Complete the following Warmup 1 problems and save them on codingbat so I can verify, comments helpful but not required
 - ix. **sleepIn, monkeyTrouble, sumDouble, diff21, parrotTrouble, makes10, icyHot, in1020, hasTeen, loneTeen, intMax, close10, in3050, max1020, lastDigit** (THERE ARE 15 HERE!!!!)
- b. Complete ALL the Logic 1 problems (THERE ARE 30 HERE!!!!)
- c. Complete ALL the Logic 2 problems (THERE ARE 9 HERE!!!!)

You can do it, **don't copy the answers from the web**, struggle through and email me if you are having grande troubles, hanlchri@shenschools.org



Project Name	Summer Assignment
Class 1 Name	TriangleCalculator
Class 2 Name	PerfectNumAnalyzer
Class 3 Name	Craps
Class 4 Name	BounceCalculator

   RUBRIC   	
TriangleCalculator	25
PerfectNumFinder	25
Craps	25
BounceCalculator	25
Presentation	30
Coding Bat (54 problems)	100
Comments	10
TOTAL	240

None, that is a hardware problem!!!!!!!!!!!!!!
See you in the fall



TENTATIVE DUE DATES

Warmup 1 from Coding bat: 15 logic problems
Each one will take approximately 10 minutes X 15 = 150 minutes.(2.5 hours)

sleepIn, monkeyTrouble, sumDouble, diff21, parrotTrouble, makes10, icyHot, in1020, hasteen, loneTeen, intMax, close10, in3050, max1020, lastDigit

Get these finished by July 15th, 6:00 PM

Logic 1 from Coding bat: 30 logic problems

Each one will take approximately 10 minutes X 30 = 300 minutes.(5 hours)

cigarParty, dateFashion, squirrelPlay, caughtSpeeding, sortaSum
alarmClock, love6 , in1To10, specialEleven, more20, old35, less20, nearTen
teenSum, answerCell, teaParty, fizzString, fizzString2, twoAsOne, inOrder
inOrderEqual, lastDigit, lessBy10, withoutDoubles, maxMod5, redTicket
greenTicket, blueTicket, shareDigit, sumLimit

Get these finished by July 31st, 6:00 PM

Logic 2 from Coding bat: 9 logic problems

makeBricks, loneSum, luckySum, noTeenSum, roundSum, closeFar, blackjack,
evenlySpaced, makeChocolate

Each one will take approximately 15 minutes X 9 = 135 minutes(2 hours, 15 min)

Get these finished by August 11th, 6:00 PM

Part 2: Google Slides Presentation, (2 hours). Share this out to
hanlchri@shenschools.org

Get these finished by August 18th, 6:00 PM

Part 3:

4 different java programs written in NetBeans

Triangle Finder 3 hours

Perfect Number(must do 3 separate methods) 3 hours

Bounce Calculator 6 hours

Craps 9 hours

21 hours total

Get these finished by August 31st, 6:00 PM

