Potential Test Questions

Below you will find potential Test Questions. Test questions might also come from codingbat.com

Grader

Level 1

Name: isItPassing
Input: double percentGrade
Output: boolean passing
Action: takes in a percentage grade and returns whether or not the student is passing. Anything at or above 60 is passing.

Level 2

Name: isError
Input: double percentGrade
Output: boolean isInError
Action: takes in a percentage that a student has in a class and returns whether the grade is an error. Any grade below 0 or above 100 is in error.

Level 3

Name: getLetter
Input: double percentGrade
Output: String letterGrade
Action: takes in a percentage that a student has in a class and returns the corresponding letter Grade that the student should receive for the class using the following scale:
[0 – 60) F, [60 – 70) D, [70 – 80) C, [80 – 90) B, [90 – 100] A

Level 4

Name: getModifier
Input: double percentGrade
Output: String modifier
Action: takes in a percentage that a student has in a class and returns the modifier that should be appended to the letter. Any percent that ends in a 7, 8, or 9 should return a "+" unless it is under 60. Any percent that ends in a 0, 1, or 2 should return a "–" unless it is less than 60 or equal to 100. A 100 should return a "+". Anything else should return an empty String "".

Level 5

Name: adjustForHonors
Input: double percentGrade
Output: double adjustedPercent
Action: takes in a percentage that a student has in a class and returns an adjusted score. The percent returned should be 10 percent higher but capped at 100. For example, with isHonors true, an 88 becomes a 98 and a 92 becomes a 100. If the percentGrade is less than 60 then the grade does not change so a 48 stays at a 48.

Name: adjustForCurve
Input: double percentGrade
Output: double adjustedPercent
Action: takes in a percentage score. It returns an adjusted score so that the number of points from 100 is multiplied by .75. So a 60% would become a 70% (-40 * .75 = -30) and a 75% would become an 81.25% (-25 * .75 = -18.75).

Level 6

Name: giveLetterGrade
Input: double percentGrade, boolean isHonors, boolean isPassFail, boolean isCurved
Output: String letterGrade
Action: takes in a percentage that a student has in a class and returns the corresponding letter Grade that the student should receive for the class. For any percentGrade that isError indicates is in error should return "Error". Any pass/fail class should return a "P" or an "F" depending on whether or not ***isItPassing*** says it is passing. Otherwise modify it first by using ***adjustForCurve***. Then modify it using ***adjustForHonors***. Then return the combination of the two strings returned from ***getLetter*** and ***getModifier***.

Cop

Level 1

Name: areTheySpeeding
Input: double speed, double speedLimit
Output: boolean speeding
Action: Returns whether or not they are speeding. Anything speed above the limit indicates speeeding

Level 2

Name: isItRecklessDriving
Input: double speed, double speedLimit
Output: boolean isReckless
Action: Returns whether the ticket is reckless driving. It is reckless driving if the speed is 20 mph above the speedLimit.

Level 3

Name: whatsTheFine
Input: double speed, double speedLimit
Output: double cost
Action: Returns how much the speeding ticket will be according to the following:

- Not speeding – 0
- Speeding but not reckless - 120 plus 10 for each mph above the limit
- Reckless – 5,000
- use ***areTheySpeeding*** to determine whether they are speeding
- use ***isItRecklessDriving*** to determine whether it is reckless

NumberAnalyzer

Level 1

Name: isItNegative
Input: int number
Output: boolean isNegative
Action: Returns whether the number sent in is negative

Level 2

Name: isItASingleDigit
Input: int number
Output: boolean isSingleDigit
Action: Returns whether the number passed in only consists of a single digit.

Level 3

Name: isItOdd
Input: int number
Output: boolean isOdd
Action: Returns whether the number sent in is odd

Name: allTheSame
Input: int num1, int num2, int num3
Output: boolean allTheSame
Action: Returns whether the three numbers passed in are all the same

Level 4

Name: whichIsLargest
Input: int num1, int num2, int num3
Output: int largestNumber
Action: Returns the largest of the three numbers passed it. If there is a tie for largest then it just returns that value.

Level 5

Name: howManyAreTheSame
Input: int num1, int num2, int num3, int num4
Output: int numSame
Action: Returns how many numbers are the same. (0, 2, or 3, 4) If there are two sets of pairs then it returns only a 2.

Level 6

Name: getMostCommonNumber
Input: int num1, int num2, int num3, int num4, int num5
Output: int commonNumber
Action: Returns the number that shows up the most. If there is a tie, then return the average of the most common numbers.

PayRoll

Level 1

Name: getBonus
Input: double sales
Output: double bonus
Action: Returns the amount the employee gets as a bonus. If sales is below 300,000 then they do not get a bonus. If their salary is above 300,000 then receive 10% of whatever amount is above 300,000. For example, sales of 500,000 would result in a bonus of 20,000 since they their sales went 200,000 above 300,000 and 10% of 200,000 is 20,000.

Level 2

Name: whatIsMyBaseSalary
Input: String degree
Output: double salary
Action: Returns the base salary of the employee given the following chart:

Doctorate Base Pay	100000
Masters Base Pay	60000
Bachelors Base Pay	40000
Associates Base Pay	25000
No Degree Base Pay	15000

Level 3

Name: whatIsMySalary
Input: String degree, int yearsExperience, double sales
Output: double salary
Action: Returns the salary of the employee. The salary should be equal to the base pay plus their bonus plus 5% of the base pay for each year of experience that the employee has. So an employee with a "Bachelors" with 10 years of experience and 500,000 in sales should have a salary of 70,000 (40,000 + 2,000 * 10 + 20,000).

Level 4

Name: taxAmountNoScale
Input: double money
Output: double - the amount of tax that the person has to pay on their income
Action: Takes in the person's income and returns the amount that they get charged in taxes. Assume that they get charged the same tax rate on all of their income.
Use the 2014 tax brackets for a single filer.

10%	$0 to $9,075
15%	$9,076 to $36,900
25%	$36,901 to $89,350
28%	$89,351 to $186,350
33%	$186,351to $405,100
35%	$405,101 to $406,750
39.60%	$406,751+

Level 5

Name: taxAmount
Input: double money
Output: double - the amount of tax that the person has to pay on their income
Action: Takes in the person's income and returns the amount that they get charged in taxes. Do not assume that they get charged the same tax rate on all of their income. Tax rates should follow a step wise pattern as they do in real life.
Use the 2014 tax brackets for a single filer.

Name: whatIsMySalaryAfterTax
Input: String degree, int yearsExperience, double sales
Output: double salary
Action: Use the whatIsMySalary method and taxAmount method to determine what the employees salary is after tax.

GoldiLocks

Level 1

Name: doYouEnterHouse
Input: boolean doorUnlocked
Output: boolean enterHouse
Action: Returns whether Goldilocks entered the house. She enters the house if the door is unlocked.

Level 2

Name: howsThePorridge
Input: double temperature
Output: boolean justRight
Action: Returns whether the porridge is the right temperature. To be the right temperature, porridge needs to be between 75 and 99 inclusive.

Name: howsTheChair
Input: char size
Output: boolean justRight
Action: Returns whether the size of the chair is just right. The method takes in a char variable size that will store a 'S' to represent small, 'M' to represent medium, or 'L' to represent large. To be just right, a chair must be medium.

Name: howsTheBed
Input: int stiffness
Output: boolean justRight
Action: Returns whether the stiffness of the bed is just right. The method takes in an int variable stiffness that will store how stiff the bed is. Positive numbers indicate that the bed is stiff. Negative numbers indicate that the bed is soft. To be just right, a bed has to be neither stiff nor soft.

Level 3

Name: howsEverything
Input: double temperature, char size, int stiffness
Output: boolean justRight
Action: Returns whether everything is justRight. To be the right temperature, porridge needs to be between 75 and 99 inclusive. To be just right, a chair must be medium ('M'). To be just right, a bed needs to be neither stiff nor soft (0).

Level 4

Name: whichPorridgeToEat
Input: double temp1, double temp2, double temp3
Output: int porridge
Action: Returns which porridge Goldilocks should eat (1, 2, or 3). In order for porridge to be acceptable, it needs to be between 75 and 99 inclusive. 90 is the best temperature though. Return which temperature (1, 2, or 3) comes closest to 90 degrees. If no temperature falls within the acceptable range, then return a -1. If two porridges tie then return the lower of the two numbers.

Name: whichChairToSitIn
Input: char size1, char size2, char size3
Output: int chair
Action: Returns which chair Goldilocks should sit in (1, 2, or 3). Goldilocks would prefer to sit in the first chair that is a medium chair ('M'). But if there are no medium chairs then Goldilocks would prefer to sit in the first large chair ('L'). If there are no medium or large chairs then Goldilocks will sit on the ground. If goldilocks sits on the ground, indicate this by returning a -1.

Name: whichBedToSleepIn
Input: int stiffness1, int stiffness2, int stiffness3
Output: int bed
Action: Returns which bed Goldilocks should sleep in (1, 2, or 3). Goldilocks will not sleep in a bed that is too stiff or to soft. Goldilocks would prefer to sleep in a bed that is just right (stiffness is 0). But given a choice between a stiff bed (stiffness is positive) and a soft bed (stiffness is negative). Given a choice between beds that are both soft or both stiff, Goldilocks would prefer to sleep in the one that is closer to just right. If there are no suitable beds then return a -1 to indicate Goldilocks is sleeping on the floor.

Academic Advisor

Level 4
Name: determineSchool
Input: int age
Output: String age
Action: Takes in the age of the student and returns the appropriate type of school according to the following:

- "Day Care" – below 5 years of age
- "Elementary School" – between 5 and 10 inclusive
- "Middle School" – between 11 and 13 inclusive
- "High School" – between 14 and 17 inclusive
- "College" – 18 and above

Level 5
Name: determineCollege
Input: double GPA, double SAT, int numHonorsClasses
Output: int tier
Action: Takes in the student's gpa, sat score, and the number of honors classes they have taken and returns which tier of college to go to using the following:

- Tier 1 - min 3.8 GPA, min 2150 SAT score, min 10 Honors Classes
- Tier 2 - min 3.4 GPA, min 1900 SAT score, min 7 Honors Classes
- Tier 3 - min 3.0 GPA, min 1750 SAT score, min 4 Honors Classes
- Tier 4 - min 2.6 GPA, min 1600 SAT score, min 2 Honors Classes
- Tier 5 - min 2.2 GPA, min 1450 SAT score, min 0 Honors Classes

Level 6
Name: determineFocus
Input: double GPA, double SAT, int numHonorsClasses
Output: String focus
Action: Takes in the student's gpa, sat score, and the number of honors classes they have taken and returns which of the three things they should focus on ("GPA", "SAT", or "Classes") to get into a higher tier school. For example, a student with a 3.5 GPA, 1620 SAT score, and 5 honors classes should focus on "SAT". A student with a 3.1 GPA, 2000 SAT, and 5 honors classes should focus on "GPA and Classes". A student with a 3.1 GPA, 1800 SAT, and 5 honors classes should focus on "GPA and SAT and Classes".

- Tier 1 - min 3.8 GPA, min 2150 SAT score, min 10 Honors Classes
- Tier 2 - min 3.4 GPA, min 1900 SAT score, min 7 Honors Classes
- Tier 3 - min 3.0 GPA, min 1750 SAT score, min 4 Honors Classes
- Tier 4 - min 2.6 GPA, min 1600 SAT score, min 2 Honors Classes
- Tier 5 - min 2.2 GPA, min 1450 SAT score, min 0 Honors Classes

Quiz

Level 1

Name: isCorrect
Input: char userAnswer, char correctAnswer
Output: boolean isCorrect
Action: Returns whether the letter passed in matches the correct answer

Name: isCorrect
Input: String userAnswer, String correctAnswer
Output: boolean isCorrect
Action: Returns whether the String passed in matches the correct answer

Level 2

Name: whichIsCorrect
Input: String answer1, String answer2, String answer3, String correctAnswer
Output: int correctAnswer
Action: Returns which of the three answers (1, 2, or 3) are correct. You may assume that no two of the answers are the same. Return a -1 if none of the answers are correct.

Level 3

Name: howManyCorrect
Input: String answer1, String answer2, String answer3, String correctAnswer
Output: int correctAnswer
Action: Returns how many of the three answers (0, 1, 2, or 3) are correct.

Name: howManyIncorrect
Input: String answer1, String answer2, String answer3, String correctAnswer
Output: int correctAnswer
Action: Returns how many of the three answers (0, 1, 2, or 3) are incorrect.

Calender

Level 4

Name: isLate
Input: String monthSubmitted, int daySubmitted, String monthDue, int dayDue
Output: String month
Action: Returns whether something is late given the month and day it was submitted and the month and day it was due.

Name: whatMonth
Input: int day
Output: String month
Action: Returns the month given the day of the year. For example, 70 returns "March".

Name: howManyDays
Input: String month
Output: int days
Action: Returns the number of days in the month. You may assume that February has 28.

Name: whichQuarter
Input: String month
Output: int quarter
Action: Returns the fiscal quarter given the month. The first fiscal quarter is October, November, December. The second is January, February, March. Third is April, May, June. Fourth is July, August, September.

Name: whatSeason
Input: String month, int day
Output: String season
Action: Returns what season it is (winter, spring, summer, or fall). Use the following for the start of the seasons:
spring - March 19, summer - June 20, fall - September 22, winter - December 21

Name: whatZodiacSign
Input: String month, int day
Output: String zodiacSign
Action: Returns what the zodiac sign is for the given month and day. Use the following chart to determine what zodiac sign it is:

Aquarius: January 20 - February 18
Pisces: February 19 - March 20
Aries: March 21 - April 19
Taurus: April 20 - May 20
Gemini: May 21 - June 20
Cancer: June 21 - July 22
Leo: July 23 - August 22

Virgo: August 23 - September 22
Libra: September 23 - October 22
Scorpio: October 23 - November 21
Sagittarius: November 22 - December 21
Capricorn: December 22 - January 19

Level 5

Name: isALeapYear
Input: int year
Output: boolean isALeapYear
Action: Returns whether the given year is a leap year. Note – leap years are not always every four year.

Name: whatDayOfTheWeekIn2019
Input: String month, int day
Output: String dayOfTheWeek
Action: Returns what day of the week it is ("Sunday" – "Saturday") given the month and day in the month and it is the year 2019.

Level 6

Name: whoIsThePresident
Input: int year, String month, int day
Output: String presidentsName
Action: Returns the who the president was on that day. Return the president's first name and last name separated by a space ("Franklin Roosevelt"). For the second John Adams and the second George Bush, return "John Adams 2" and "George Bush 2"

Name: whatDayOfTheWeek
Input: String month, int day
Output: String dayOfTheWeek
Action: Returns what day of the week it is ("Sunday" – "Saturday") given the month and day in the month.

GeoAnalyzer

Level 4

Name: isInBounds
Input: double value, double min, double max
Output: boolean inBounds
Action: Returns whether the value is between min and max inclusive

Name: isOutOfBounds
Input: double value, double min, double max
Output: boolean outOfBounds
Action: Returns whether the value is outside of min and max exclusive

Name: couldFitInBounds
Input: double length, double min, double max
Output: boolean couldFit
Action: Returns whether the value could fit between min and max inclusive

Name: isItEnoughGas
Input: double gasAmount, double mpg, double distance
Output: boolean enoughGas
Action: Returns whether the amountOfGas passed in is enough to make it the distance indicated.

Name: pointInBounds
Input: double x, double y, double xMin, double xMax, double yMin, double yMax
Output: boolean inBounds
Action: Returns whether the (x,y) values are inside the indicated bounds

Name: pointOutOfBounds
Input: double x, double y, double xMin, double xMax, double yMin, double yMax
Output: boolean outOfBounds
Action: Returns whether the (x,y) values are outside the indicated bounds

Name: couldFitInBounds
Input: double width, double height, double xMin, double xMax, double yMin, double yMax
Output: boolean couldFit
Action: Returns whether a box with the indicated width and height could fit within the indicated bounds

Name: allOnLine
Input: double x1, double y1, double x2, double y2, double x3, double y3
Output: boolean allOnLine
Action: Returns whether the three points are all on the same line

Name: isSameRatio
Input: double numerator1, double denominator1, double numerator2, double denominator2
Output: boolean sameRatio
Action: Returns whether the two ratios passed in are the same

Name: isRightTriangle
Input: double leg1, double leg2, double hypotenuse
Output: boolean isRightTriangle
Action: Returns the whether the triangle is a right triangle given the length of the two legs and the hypotenuse.

Name: whatTypeOfTriangle
Input: double leg1, double leg2, double hypotenuse
Output: String triangleType
Action: Returns the type of triangle given the three side lengths. Return “right”, “acute”, or “obtuse”

Name: isTriangle
Input: double side1, double side2, double side3
Output: boolean isTriangle
Action: Returns whether a triangle can be composed of the three side lengths.

Name: whatTypeOfParallelogram
Input: double side1, double side2, double side3, double side4, double angle1, double angle2, double angle3, double angle4
Output: String quadType
Action: Returns the type of quadrilateral given the side lengths and angle measure. Return the most accurate description (“quadrilateral”, “parallelogram”, “rectangle”, “rhombus”, “square”, “kite”, “trapezoid”, “isosceles trapezoid”)

Name: areLinesParallelGivenAltInteriorAngles
Input: double altIntAngle1, double altIntAngle2
Output: boolean areParallel
Action: Returns whether the lines are parallel given the measure of two alternate interior angles

Name: areLinesParallelGivenConsecutiveInteriorAngles
Input: double consIntAngle1, double consIntAngle2
Output: boolean areParallel
Action: Returns whether the lines are parallel given the measure of two consecutive interior angles

Name: areLinesRelated
Input: double slope1, double slope2
Output: String relation
Action: Returns whether the lines are “parallel”, “perpendicular”, given the measure of the two slopes

LegalEagle

Name: qualifyForSocialSecurity
Input: int age
Output: boolean getSocialSecurity
Action: Returns whether the person qualifies for Social Security. To qualify someone must be 67.

Name: canRunForPresident
Input: int age
Output: boolean canRunForPresident
Action: Returns whether the person can be president. To be president, they must be 35 or older.

Name: canYouEnlistInArmy
Input: int age
Output: boolean canEnlist
Action: Returns whether the person can enlist in the army. To enlist, someone must be at least 18 and at most 35 years of age.

Name: qualifyToVote
Input: int age, boolean isFelon
Output: boolean canVote
Action: Returns whether the person is qualified. To qualify to vote, someone must be 18 year of age and have not been convicted of a felony.

Other

Name: isNormalTemperature
Input: double temp
Output: boolean isNormal
Action: Returns whether the temperature passed in is a normal temp. A normal temperature should be 98.6 degrees.

Name: isItEnough
Input: double money, double price
Output: boolean isEnough
Action: Returns whether the person has enough money to purchase the item given the amount of money they have and the price of the item.

Name: whoIsTheWinner
Input: double time1, String name1, double time2, String name2
Output: String name
Action: Returns which runner is the winner depending on their times. If the two times are the same then return "tie"

Name: scrimmageForHowLong
Input: boolean isRaining, double temp
Output: double time
Action: Returns how long to scrimmage for. On a regular day you scrimmage for 60 minutes. When it is a hot or cold day, you scrimmage for only 30 minutes. Any time it is raining though, you do not scrimmage at all (0 minutes).

Name: whatsTheGreeting
Input: int age
Output: String greeting
Action: Returns the greeting that should is to be said given someone's age:

- 0-3: A googo gaga
- 4-7: Let me pinch those checks
- 8-12: You are just adorable
- 13-18: Yo
- 19-20: What are you up to these days?
- 21-29: Hello
- 30-55: How's work?
- 56-65: A good day sir
- 66-80: How's retirement
- 81&up: Ahoho

Name: getTruckSize
Input: int numCouches, int numChairs, int numTables
Output: String truckType
Action: Returns which truck should be used to move the furniture. Use the following list for dimensions. If there is no truck big enough then return "load to large".

- couch - 250 cubic feet
- chair - 20 cubic feet
- table - 80 cubic feet
- "little" truck - 200 cubic feet
- "medium" truck - 600 cubic feet
- "large" truck - 2000 cubic feet

Name: getWinnings
Input: int slot1, int slot2, int slot3
Output: int winnings
Action: Returns the amount that was won. If all three values are the same, then return the value multiplied by 100. If only two of the values are the same, then return the value multiplied by 10.

Name: getThief
Input: int timeOfVisit, boolean hasLontHair, boolean isAdult, boolean hasPaws
Output: int winnings
Action: Someone stole Mr. Mayewsky's candy. The only people that could have done it are below along with their attributes. Write the method that will tell the user who stole the candy.

Name	Time of Visit	Has Long Hair	Is Adult	Has Paws
Fred	before 3	false	true	false
Wilma	before 3	true	true	false
Pebbles	before 3	true	false	false
Dino	before 3	false	true	true
Barney	arrived at 3	false	true	false
Betty	arrived at 3	true	true	false
Bamm-Bamm	arrived at 3	false	false	false