Sample Exam Questions

The sample exam questions that follow illustrate the relationship between the course framework and the AP Computer Science A Exam and serve as examples of the types of questions that appear on the exam. These sample questions do not represent the full range and distribution of items on an official AP Computer Science A Exam. After the sample questions is a table that shows which skill, learning objective(s), and essential knowledge statement(s) each question assesses. The table also provides the answers to the multiple-choice questions.

Section I: Multiple-Choice

1. Consider the following code segment.

```
double q = 15.0;
int r = 2;

double x = (int) (q / r);
double y = q / r;

System.out.println(x + " " + y);
```

What is printed as a result of executing this code segment?

(A) `7.0 7.0`

(B) `7.0 7.5`

(C) `7.5 7.0`

(D) `7.5 7.5`

2. Consider the following code segment.

```
int x = ((int) (Math.random() * 10)) + 5;
int y = ((int) (Math.random() * 5)) + 10;
System.out.println(x + " " + y);
```

Which of the following could be printed as a result of executing this code segment?

(A) `8 15`

(B) `9 12`

(C) `10 7`

(D) `15 11`

© 2025 College Board

3. Consider the following `Date` class.

```
public class Date
{
   private String month;
   private int day;

   public Date(String m, int d)
   { /* implementation not shown */ }

   /* Class contains no other constructors.
      Other methods are not shown. */
}
```

Which of the following code segments, appearing in a class other than `Date`, will correctly create an instance of a `Date` object?

(A)
```
Date birthday = new Date("September", "5th");
```

(B)
```
Date birthday = new Date("September", 5);
```

(C)
```
Date birthday = new Date("September 5");
```

(D)
```
Date birthday = new Date();
birthday.month = "September";
birthday.day = 5;
```

4. Consider the following code segment.

```
String str1 = "LMNOP";
String str2 = str1.substring(3);
str2 += str1.substring(2, 3);
```

What is the value of `str2` after executing this code segment?

(A) `"OPN"`

(B) `"OPNO"`

(C) `"NOPM"`

(D) `"NOPMN"`

5. The following method is intended to return `"Renaissance"` for years from 1400 to 1600, inclusive, `"Medieval"` for years from 400 to 1399, inclusive, or `"Other"` for any other year. In the method, the parameter represents a year.

```
public static String getCategory(int year)
{
   String category = "";
   /* missing code */
   return category;
}
```


© 2025 College Board

Which of the following code segments can replace `/* missing code */` so that the `getCategory` method works as intended?

(A)
```
if (year > 1600 || year < 400)
{
   category = "Other";
}
else if (year >= 1400)
{
   category = "Renaissance";
}
else if (year >= 400)
{
   category = "Medieval";
}
```

(B)
```
if (year > 1600 || year < 400)
{
   category = "Other";
}
if (year >= 1400)
{
   category = "Renaissance";
}
if (year >= 400)
{
   category = "Medieval";
}
```

(C)
```
if (year >= 1400)
{
   category = "Renaissance";
}
else if (year >= 400)
{
   category = "Medieval";
}
else
{
   category = "Other";
}
```

(D)
```
if (year >= 1400)
{
   category = "Renaissance";
}
if (year >= 400)
{
   category = "Medieval";
}
else
{
   category = "Other";
}
```


© 2025 College Board

6. Assume that `x`, `y`, and `z` are `boolean` variables that have been properly declared and initialized.

   Which of the following expressions is equivalent to `!x || !y || z`?

   (A) `!(x && y) && z`

   (B) `!(x && y) || z`

   (C) `!(x || y) && z`

   (D) `!(x || y) || z`

7. Consider the following code segment.

```
for (int j = 4; j > 0; j--)
{
   for (/* missing code */)
   {
      System.out.print(k + " ");
   }
   System.out.println();
}
```

   This code segment is intended to produce the following output.

```
0 1 2 3
0 1 2
0 1
0
```

   Which of the following can be used to replace `/* missing code */` so that this code segment works as intended?

   (A) `int k = 0; k < j; k++`

   (B) `int k = 0; k <= j; k++`

   (C) `int k = j; k > 0; k--`

   (D) `int k = j; k >= 0; k--`

8. Consider the following code segment.

```
int j = 1;
while (j < 4)
{
   for (int k = 0; k <= 4; k++)
   {
      System.out.println("hello");  // line 6
   }
   j++;
}
```

   How many times will the statement in line 6 be executed as a result of running this code segment?

   (A) 12

   (B) 15

   (C) 16

   (D) 20

© 2025 College Board

9. A programmer is creating a `Movie` class that will contain attributes for a movie's title and its average user rating on a scale from `0.0` to `5.0`. The class must also contain methods to allow the attributes to be accessed outside the class.

   A class design diagram for the `Movie` class will contain three sections. The first section contains the class name, the second section contains two instance variables and their data types, and the third section contains two methods and their return types. The `+` symbol indicates a `public` designation, and the symbol `-` indicates a `private` designation.

   Which of the following diagrams represents the most appropriate design for the `Movie` class?

(A)

Movie
+ title : String + rating : double
- getTitle() : String - getRating() : double

(B)

Movie
+ title : String + rating : int
- getTitle() : String - getRating() : int

(C)

Movie
- title : String - rating : double
+ getTitle() : String + getRating() : double

(D)

Movie
- title : String - rating : int
+ getTitle() : String + getRating() : int

© 2025 College Board

10. A programmer developed a method and wants to allow other programmers to use the method in other programs without raising intellectual property concerns. Which of the following actions is most likely to support this goal?

    (A) Designating the method as `public` in the method header

    (B) Designating the method as `static` in the method header

    (C) Publishing the code for the method as open source

    (D) Removing all personal information from the data used by the method

11. The following class declarations are used to represent information about an apartment and a duplex, which is a building containing two apartments.

```
public class Apartment
{
   public int calculateRent()
   { /* implementation not shown */ }

   private String getTenant()
   { /* implementation not shown */ }

   /* There may be instance variables, methods,
      and constructors that are not shown. */
}

public class Duplex
{
   private Apartment unitOne;
   private Apartment unitTwo;

   public void printInfo()
   {
      System.out.println( /* missing code */ );
   }

   /* There may be instance variables, methods,
      and constructors that are not shown. */
}
```

    Which of the following replacements for `/* missing code */` will compile without error?

    (A) `Apartment.calculateRent()`

    (B) `Apartment.getTenant()`

    (C) `unitOne.calculateRent()`

    (D) `unitTwo.getTenant()`

© 2025 College Board

12. Consider the following class declaration.

```
public class Point
{
   private int xCoor;  // the x-coordinate of the point
   private int yCoor;  // the y-coordinate of the point

   public Point(int x, int y)
   {
      xCoor = x;
      yCoor = y;
   }

   public int getXCoor()
   {
      return xCoor;
   }

   public int getYCoor()
   {
      return yCoor;
   }
   public boolean isEquiv(Point otherP)
   {
      return /* missing expression */;
   }
}
```

Two `Point` objects are considered equivalent if they have the same *x* and *y* coordinates. The `isEquiv` method is intended to return `true` if this `Point` object is equivalent to the `Point` object parameter and return `false` otherwise.

Which of the following expressions can replace `/* missing expression */` so that the method works as intended?

(A) `getXCoor() == xCoor && getYCoor() == yCoor`

(B) `otherP.xCoor == x && otherP.yCoor == y`

(C)
```
otherP.getXCoor() == getXCoor() &&
otherP.getYCoor() == getYCoor()
```

(D) `otherP == this`

© 2025 College Board

13. A researcher wants to investigate which age group (under 18 years old, or 18 years old and over) is more likely to have downloaded a certain application from an online store. Each customer of the store has a unique ID number. The researcher has the following data sets available.

- Data set 1 contains an entry for each application available from the store. Each entry includes the name of the application and the total number of customers who have downloaded it.
- Data set 2 contains an entry for each application available from the store. Each entry includes the ID number and age of the customer who most recently downloaded the application.
- Data set 3 contains an entry for each customer of the store. Each entry includes the customer's ID number and the customer's age.
- Data set 4 contains an entry for each customer of the store. Each entry includes the customer's ID number and the names of all the applications downloaded by the customer.

Which two data sets can be combined and analyzed to determine the desired information?

(A) Data sets 1 and 2

(B) Data sets 2 and 3

(C) Data sets 2 and 4

(D) Data sets 3 and 4

14. Consider the following method, which is intended to return the maximum integer value contained in `ranges`.

```
/** Precondition: ranges contains at least one element. */
public int findMaxValue(int[] ranges)
{
   int max = 0;                 // line 4
   for (int value : ranges)
   {
      if (value > max)          // line 7
      {
         max = value;           // line 9
      }
   }
   return max;
}
```

This method does not always work as intended. Which of the following changes can be made so that the code segment works as intended?

(A) Changing line 4 to `int max = ranges[0];`

(B) Changing line 4 to `int max = Integer.MAX_VALUE;`

(C) Changing line 7 to `if (value < max)`

(D) Changing line 9 to `value = max;`

© 2025 College Board

15. A text file named `roster.txt` has the following contents. Assume that the file contains no spaces.

```
Cohen-Isabel
Lee-Cindy
Patel-Anisha
Sanchez-Michael
```

In the following code segment, a valid `Scanner` object named `scan` has been created to read from the text file.

```
File myText = new File("roster.txt");
Scanner scan = new Scanner(myText);

ArrayList<String> usernames = new ArrayList<String>();

/* missing code */
```

This code segment is intended to assign `["ICohen", "CLee", "APatel", "MSanchez"]` to `usernames`.

Which of the following can replace `/* missing code */` so that this code segment works as intended?

(A)
```
while (scan.hasNext())
   {
      String temp = scan.next().substring(0, 1);
      usernames.add(temp + scan.next());
   }
```

(B)
```
while (scan.hasNext())
   {
      String one = scan.next();
      String two = scan.next();
      usernames.add(two.substring(0, 1) + one);
   }
```

(C)
```
while (scan.hasNext())
   {
      String[] temp = scan.next().split("-");
      usernames.add(temp[1].substring(0, 1) + temp[0]);
   }
```

(D)
```
while (scan.hasNext())
   {
      String[] temp = scan.next().split("-");
      usernames.add(temp[0].substring(0, 1) + temp[1]);
   }
```


© 2025 College Board

16. Consider the following method.

```
public void reviseList(ArrayList<Integer> list)
{
   for (int i = 0; i < list.size(); i++)
   {
      int temp = list.get(i);
      if (temp % 2 == 1)
      {
         list.set(i, temp + 1);
      }
   }
}
```

The `reviseList` method is called with an `ArrayList` containing the values `[1, 2, 3, 3, 2, 1]`. What will be the contents of the `ArrayList` after executing this method?

(A) `[1, 3, 3, 3, 3, 1]`

(B) `[1, 3, 3, 4, 2, 2]`

(C) `[2, 2, 4, 4, 2, 2]`

(D) `[2, 3, 4, 4, 3, 2]`

17. Consider the following code segment.

```
int[][] arr2D = /* missing initialization */;
int sum = 0;

for (int j = 0; j < arr2D.length; j++)
{
   sum += arr2D[j][2];
}

System.out.println(sum);
```

Which of the following replacements for `/* missing initialization */` will cause the code segment to print the value `3`?

(A)
```
{{0, 0, 1},
 {0, 0, 2},
 {0, 0, 1}}
```

(B)
```
{{0, 0, 0},
 {0, 1, 2},
 {0, 0, 0}}
```

(C)
```
{{0, 0, 0},
 {0, 0, 0},
 {1, 1, 1},
 {0, 0, 0}}
```

(D)
```
{{0, 0, 1, 0},
 {0, 0, 1, 0},
 {0, 0, 1, 0}}
```


© 2025 College Board

18. Consider the following code segment.

```
int[][] arr = {{10, -7, 19, 14},
               {-3, -2, -6, 11},
               {-9, 12, 10, -1}};
int ans = 1;
for (int[] row : arr)
{
   for (int value : row)
   {
      if (value > 0)
      {
         ans = ans * value;
      }
   }
}
System.out.println(ans);
```

Which of the following best describes the behavior of the code segment?

(A) It prints the product of all the elements in `arr`.

(B) It prints the product of all the negative elements in `arr`.

(C) It prints the product of all the positive elements in `arr`.

(D) It prints nothing because of a run-time error.

© 2025 College Board

19. The following method is a correct implementation of the selection sort algorithm. The method sorts the elements of `arr` so that they are in order from least to greatest.

```
public static void selectionSort(int[] arr)
{
   for (int j = 0; j < arr.length - 1; j++)
   {
      int minIndex = j;
      for (int k = j + 1; k < arr.length; k++)
      {
         if (arr[k] < arr[minIndex])
         {
            minIndex = k;
         }
      }

      int temp = arr[j];
      arr[j] = arr[minIndex];
      arr[minIndex] = temp;
      /* end of outer loop */
   }
}
```

Assume that `selectionSort` has been called with an `int[]` argument that has been initialized with the following contents.

```
{40, 30, 50, 60, 10, 20}
```

What will the contents of `arr` be after three iterations of the outer loop (i.e., when `j == 2` at the point indicated by `/* end of outer loop */`)?

(A) `{10, 20, 30, 40, 60, 50}`

(B) `{10, 20, 30, 60, 40, 50}`

(C) `{10, 20, 50, 60, 40, 30}`

(D) `{10, 30, 50, 60, 40, 20}`

20. Consider the following method.

```
public static void mystery(int j, int k)
{
   if (j > k)
   {
      mystery(j – 2, k + 2);
   }
   System.out.print(j + " ");
}
```

What is printed as a result of the call `mystery(10, 0)`?

(A) `10 8 6 4`

(B) `10 8 6`

(C) `6 8 10`

(D) `4 6 8 10`

© 2025 College Board

Section II: Free-Response

The following are examples of the kinds of free-response questions found on the exam.

Question 1: Methods and Control Structures

This question involves the `MessageBuilder` class, which is used to generate a message based on a starting word. The `MessageBuilder` class contains a helper method, `getNextWord`. You will write a constructor and a method in the `MessageBuilder` class.

```
public class MessageBuilder
{
   private String message; // To be initialized in part (a)
   private int numWords;   // To be initialized in part (a)

   /**
    * Builds a message starting with the word specified by the
    * parameter and counts the number of words in the message,
    * as described in part (a)
    * Precondition: startingWord is a single word with no spaces.
    */
   public MessageBuilder(String startingWord)
   { /* to be implemented in part (a) */ }

   /**
    * Returns a word to follow the word specified by the
    * parameter or null if there are no remaining words.
    * Precondition: s is a single word with no spaces.
    * Postcondition: Returns an individual word with no spaces.
    */
   public String getNextWord(String s)
   { /* implementation not shown */ }

   /**
    * Returns an abbreviation for the instance variable message,
    * as described in part (b)
    * Preconditions: Each word in message is separated by a
    *                single space.
    *                message contains two or more words.
    * Postcondition: message is unchanged.
    */
   public String getAbbreviation()
   { /* to be implemented in part (b) */ }

   /* There may be instance variables, constructors, and methods
      that are not shown. */
}
```


© 2025 College Board

(a) Write the `MessageBuilder` constructor, which uses a helper method to create a message starting with the word specified by the parameter `startingWord` and assigns the message to the instance variable `message`. The constructor also counts the number of words in the message and assigns the count to the instance variable `numWords`.

Each word in the message will be separated by a single space.

A helper method, `getNextWord`, has been provided to obtain words to be added to the message. Each call to `getNextWord` returns the next word in the message based on the word most recently added to the message. When there are no more words to be added to the message, `getNextWord` returns `null`. Consecutive calls to `getNextWord` are guaranteed to eventually return `null`.

Example 1

Consider the following calls to `getNextWord`, which are made within the `MessageBuilder` class. The return value of each call is used in the next call.

Method Call	Return Value
`getNextWord("the")`	`"book"`
`getNextWord("book")`	`"on"`
`getNextWord("on")`	`"the"`
`getNextWord("the")`	`"table"`
`getNextWord("table")`	`null`

Based on these return values, a call to the `MessageBuilder` constructor with argument `"the"` should set the instance variable message to `"the book on the table"` and should set the instance variable `numWords` to `5`.

Example 2

Consider the following calls to `getNextWord`, which are made within the `MessageBuilder` class. The return value of each call is used in the next call.

Method Call	Return Value
`getNextWord("good")`	`"morning"`
`getNextWord("morning")`	`"sunshine"`
`getNextWord("sunshine")`	`null`

Based on these return values, a call to the `MessageBuilder` constructor with argument `"good"` should set the instance variable `message` to `"good morning sunshine"` and should set the instance variable `numWords` to `3`.

© 2025 College Board

Complete the `MessageBuilder` constructor. You must use `getNextWord` appropriately in order to receive full credit.

```
/**
 * Builds a message starting with the word specified by the
 * parameter and counts the number of words in the message,
 * as described in part (a)
 * Precondition: startingWord is a single word with no spaces.
 */
public MessageBuilder(String startingWord)
```

(b) Write the `getAbbreviation` method, which returns an abbreviation consisting of the first letter of each word in `message`. Assume that `message` consists of two or more words, each separated by a single space.

For example, if the value of `message` is `"as soon as possible"`, then `getAbbreviation()` should return `"asap"`.

Complete the `getAbbreviation` method.

```
/**
 * Returns an abbreviation for the instance variable message,
 * as described in part (b)
 * Preconditions: Each word in message is separated by a
 *                single space.
 *                message contains two or more words.
 * Postcondition: message is unchanged.
 */
public String getAbbreviation()
```


© 2025 College Board

Question 2: Class Design

The `CupcakeMachine` class, which you will write, represents a cupcake vending machine, an automated machine that dispenses cupcakes. `CupcakeMachine` objects are created by calls to a constructor with two parameters.

- The first parameter is an `int` that represents the number of cupcakes that the vending machine has been stocked with. Assume that this value will be greater than or equal to `0`.
- The second parameter is a `double` that represents the cost, in dollars, per cupcake. Assume that this value will be greater than `0.0`.

The `CupcakeMachine` class contains a `takeOrder` method, which determines whether a cupcake order can be filled. A cupcake order can be filled if there are at least as many cupcakes in the vending machine as there are in the order.

A cupcake order is represented by a single `int` parameter to the `takeOrder` method. Assume that all values passed to the `takeOrder` method are positive.

If an order can be filled, the method updates the number of cupcakes remaining in the machine and returns a `String` containing information about the order. The returned string indicates the order number and the cost of the order, as shown in the following table. Order numbers begin at `1` and increase by `1` for each order filled.

If the order cannot be filled because the vending machine does not have enough cupcakes, the `takeOrder` method should return the message `"Order cannot be filled"`. In this case, the number of cupcakes available in the machine is unchanged and no order number is given to the order.

The following table contains a sample code execution sequence and the corresponding results. The code execution sequence appears in a class other than `CupcakeMachine`.

© 2025 College Board

Statement	Return Value (blank if no value)	Explanation
`String info;`		
`CupcakeMachine c1 = new CupcakeMachine(10, 1.75);`		`CupcakeMachine c1` is constructed with 10 cupcakes and a cost of $1.75 per cupcake.
`info = c1.takeOrder(2);`	`"Order number 1, cost $3.5"`	A customer orders 2 cupcakes for a total cost of 2 × $1.75. There are now 8 cupcakes remaining in the machine.
`info = c1.takeOrder(3);`	`"Order number 2, cost $5.25"`	A customer orders 3 cupcakes for a total cost of 3 × $1.75. There are now 5 cupcakes remaining in the machine.
`info = c1.takeOrder(10);`	`"Order cannot be filled"`	A customer attempts to order 10 cupcakes. The machine only has 5 cupcakes available, so no order is made.
`info = c1.takeOrder(1);`	`"Order number 3, cost $1.75"`	A customer orders 1 cupcake for a total cost of $1.75. There are now 4 cupcakes remaining in the machine.
`CupcakeMachine c2 = new CupcakeMachine(10, 1.5);`		`CupcakeMachine c2` is constructed with 10 cupcakes and a cost of $1.50 per cupcake.
`info = c2.takeOrder(10);`	`"Order number 1, cost $15.0"`	A customer orders 10 cupcakes for a total cost of 10 × $1.50. There are now 0 cupcakes remaining in the machine.

Write the complete `CupcakeMachine` class. Your implementation must meet all specifications and conform to the examples shown in the table.

© 2025 College Board

Question 3: Data Analysis with `ArrayList`

The `ItemInfo` class is used to store information about an item at a store. A partial declaration of the `ItemInfo` class is shown.

```
public class ItemInfo
{
   /**
    * Returns the name of the item
    */
   public String getName()
   { /* implementation not shown */ }

   /**
    * Returns a value greater than 0.0 that represents the
    * cost of a single unit of the item, in dollars
    */
   public double getCost()
   { /* implementation not shown */ }

   /**
    * Returns true if the item is currently available and
    * returns false otherwise
    */
   public boolean isAvailable()
   { /* implementation not shown */ }

   /* There may be instance variables, constructors, and
      methods that are not shown. */
}
```


© 2025 College Board

The `ItemInventory` class maintains an `ArrayList` named `inventory` that contains all items at the store. A partial declaration of the `ItemInventory` class is shown.

```
public class ItemInventory
{
   /** The list of all items at the store */
   private ArrayList<ItemInfo> inventory;

   /**
    * Returns the average cost of the available items
    * whose cost is between lower and upper, inclusive
    * Precondition: lower <= upper
    *               At least one available element of
    *               inventory has a cost between
    *               lower and upper, inclusive.
    *               No elements of inventory are null.
    */
   public double averageWithinRange(double lower, double upper)
   { /* to be implemented */}

   /* There may be instance variables, constructors, and methods
      that are not shown. */
}
```

Write the `ItemInventory` method `averageWithinRange`. The method should return the average cost of the available items in `inventory` whose cost is between the parameters `lower` and `upper`, inclusive.

Suppose `inventory` contains the following seven `ItemInfo` objects.

Name	`"action figure"`	`"hair brush"`	`"frying pan"`	`"dish sponge"`	`"coffee mug"`	`"scarf"`	`"watch"`
Cost	`20.0`	`7.99`	`45.0`	`2.0`	`10.0`	`59.0`	`45.0`
Is Available	`true`	`true`	`true`	`false`	`true`	`true`	`false`

For the inventory shown, `averageWithinRange(10.0, 50.0)` should return `25.0`, which is equal to the average cost of the available items within the specified range (a $20 action figure, a $45 frying pan, and a $10 coffee mug). Although the watch is within the specified range, it is not available.

Complete method `averageWithinRange`.

```
/**
 * Returns the average cost of the available items
 * whose cost is between lower and upper, inclusive
 * Precondition: lower <= upper
 *               At least one available element of
 *               inventory has a cost between
 *               lower and upper, inclusive.
 *               No elements of inventory are null.
 */
public double averageWithinRange(double lower, double upper)
```


© 2025 College Board

Question 4: 2D Array

The `Appointment` class is used to store information about a person's scheduled appointments. A partial declaration of the `Appointment` class is shown.

```
public class Appointment
{
   /**
    * Returns the status of the appointment ("free", "busy", etc.)
    */
   public String getStatus()
   { /* implementation not shown */ }

   /**
    * Returns the room number of the appointment location
    */
   public int getRoomNumber()
   { /* implementation not shown */ }

   /* There may be instance variables, constructors, and methods
      that are not shown. */
}
```

The `Schedule` class maintains a two-dimensional array of appointments that represents a person's schedule. A partial declaration of the `Schedule` class is shown.

```
public class Schedule
{
   private Appointment[][] sched;

   /**
    * Returns the index of a column containing the fewest
    * occurrences of the status indicated by the parameter
    * target
    * Preconditions: sched is not null and no elements
    *                of sched are null.
    *                sched has at least one row and at
    *                least one column.
    */
   public int columnWithFewest(String target)
   { /* to be implemented */}

   /* There may be instance variables, constructors, and methods
      that are not shown. */
}
```

When an element of the two-dimensional array `sched` is accessed, the first index is used to specify the row and the second index is used to specify the column.

© 2025 College Board

Write the `Schedule` method `columnWithFewest`. The method should return the index of a column in `sched` that contains the fewest occurrences of the parameter `target`. If there are multiple columns that have the fewest number of occurrences of `target`, any of their column indices can be returned.

Suppose `sched` has the following contents. For each element, the first value is the appointment status and the second value is the room number.

	0	1	2	3	4
0	"free" 100	"free" 100	"free" 100	"busy" 206	"busy" 204
1	"free" 100	"free" 100	"busy" 304	"busy" 206	"busy" 202
2	"hold" 201	"busy" 105	"busy" 205	"free" 100	"busy" 205
3	"busy" 204	"free" 100	"busy" 310	"hold" 110	"free" 100
4	"busy" 204	"hold" 201	"hold" 310	"busy" 105	"free" 100
5	"busy" 105	"busy" 208	"hold" 310	"busy" 105	"free" 100

For these contents of `sched`, the expected behavior of `columnWithFewest` is as follows.

- The call `columnWithFewest("busy")` should return `1` because `"busy"` appears two times in column `1` and more than two times in each of the other columns.
- The call `columnWithFewest("free")` should return either `2` or `3` because `("free")` appears one time in column `2`, one time in column `3`, and more than one time in each of the other columns.
- The call `columnWithFewest("hold")` should return `4` because `"hold"` appears zero times in column `4` and one or more times in each of the other columns.

Complete method `columnWithFewest`.

```
/**
 * Returns the index of a column containing the fewest
 * occurrences of the status indicated by the parameter
 * target
 * Preconditions: sched is not null and no elements
 *                of sched are null.
 *                sched has at least one row and at
 *                least one column.
 */
public int columnWithFewest(String target)
```


© 2025 College Board