

Practice Exam One

COMPUTER SCIENCE A

SECTION I

Time—1 hour and 15 minutes

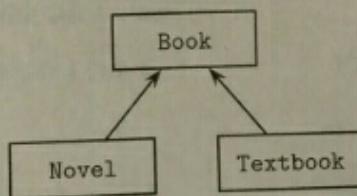
Number of questions—40

Percent of total grade—50

Directions: Determine the answer to each of the following questions or incomplete statements, using separate pieces of scrap paper for any necessary scratchwork. Then decide which is the best of the choices given and fill in the corresponding oval on the answer sheet. Do not spend too much time on any one problem.

Note: Assume that the standard packages (e.g., `java.util.*`) are included in any programs that use the code segments provided in individual questions. A Quick Reference to the standard classes and interfaces with their required methods is provided.

1. Consider this inheritance hierarchy, in which `Novel` and `Textbook` are subclasses of `Book`.



Which of the following is a *false* statement about the classes shown?

- (A) The `Textbook` class can have private instance variables that are neither in `Book` nor `Novel`.
- (B) Each of the classes—`Book`, `Novel`, and `Textbook`—can have a method `computeShelfLife`, whose code in `Book` and `Novel` is identical, but different from the code in `Textbook`.
- (C) If the `Book` class has private instance variables `myTitle` and `myAuthor`, then `Novel` and `Textbook` inherit them but cannot directly access them.
- (D) Both `Novel` and `Textbook` inherit the constructors in `Book`.
- (E) If the `Book` class has a static method called `getPublisher`, this method may not be overridden in either the `Novel` or `Textbook` classes.

2. A programmer is designing a program to catalog all books in a library. He plans to have a `Book` class that stores features of each book: `author`, `title`, `isOnShelf`, and so on, with operations like `getAuthor`, `getTitle`, `getShelfInfo`, and `setShelfInfo`. Another class, `LibraryList`, will store an array of `Book` objects. The `LibraryList` class will include operations such as `listAllBooks`, `searchForBook`, `removeBook`, and `addBook`. The programmer plans to implement and test the `Book` class first, before implementing the `LibraryList` class. The programmer's plan for writing this program is an example of
- (A) Top-down development.
 - (B) Bottom-up development.
 - (C) Stepwise refinement.
 - (D) Implementing an interface.
 - (E) A driver program.

Questions 3–6 refer to the `Card` class shown below.

```
public class Card
{
    private String mySuit;
    private int myValue;      //0 to 12

    public Card(String suit, int value)
    { implementation }

    public String getSuit()
    { return mySuit; }

    public int getValue()
    { return myValue; }

    public String toString()
    {
        String faceValue = "";
        if (myValue == 11)
            faceValue = "J";
        else if (myValue == 12)
            faceValue = "Q";
        else if (myValue == 0)
            faceValue = "K";
        else if (myValue == 1)
            faceValue = "A";
        if (myValue >= 2 && myValue <= 10)
            return myValue + " of " + mySuit;
        else
            return faceValue + " of " + mySuit;
    }
}
```

3. Which is *true* about the methods of the Card class?
- The Card class has a default constructor.
 - The constructor with parameters allows the suit and value for a Card to be initialized from user input.
 - The `toString` method returns a deck of Card objects as a string.
 - The Card class has a default method that allows comparison of Card objects.
 - The `equals` method can be used to test whether two Card objects have the same suit and value.

4. Which of the following represents correct *implementation* code for the constructor in the Card class?

- `mySuit = suit;`
`myValue = value;`
- `suit = mySuit;`
`value = myValue;`
- `Card = new Card(mySuit, myValue);`
- `Card = new Card(suit, value);`
- `mySuit = getSuit();`
`myValue = getValue();`

5. A programmer is designing a program that will play a "pick-a-card" game with a single deck of cards. Eventually the player with the highest card wins. If the programmer plans to use the Card class in the program, which of the following is the best modification to compare two cards for greater than or less than?

I Modify the Card class header as follows:

```
public class Card implements Comparable
```

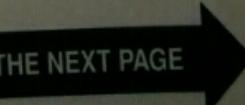
II Add a `compareTo` method to the Card class.

III Add an `equals` method to the Card class.

- I only
- II only
- III only
- I and II only
- I, II, and III

6. Which of the following correctly constructs a Card object?

- `Card c = Card("spades", 10);`
- `Card c = new Card("clubs", "J");`
- `Card c;`
`c = Card("spades", 4);`
- `Card c = new Card("hearts", "8");`
- `Card c = new Card("diamonds", 12);`

GO ON TO THE NEXT PAGE 

To simulate a deck of cards, the following Deck class will make use of the Card class for Questions 7–9.

```
public class Deck
{
    private Card[] myDeck;
    public final static int NUMCARDS = 52;

    public Deck()
    { implementation }

    //Simulate shuffling the deck
    public void shuffle()
    { implementation }

    //other methods
    ...
}
```

Refer to this description of the Deck constructor for Questions 7 and 8.

A Deck object will be constructed as follows:

myDeck[0]...myDeck[12] will contain the spade suit
 myDeck[13]...myDeck[25] will contain the heart suit
 myDeck[26]...myDeck[38] will contain the diamond suit
 myDeck[39]...myDeck[51] will contain the club suit

In each suit the card values range from 0 to 12. (These are converted to actual card values in the `toString` method of the Card class.) Here is the constructor for the Deck class:

```
public Deck()
{
    < declaration of the myDeck array >

    for (int i=0; i<NUMCARDS; i++)
    {
        < code to insert the spade cards into myDeck >
        < code to insert the heart cards into myDeck >
        < code to insert the diamond cards into myDeck >
        < code to insert the club cards into myDeck >
    }
}
```

7. Which of the following is a correct < declaration of the myDeck array >?
- `myDeck = new Deck[NUMCARDS];`
 - `myDeck = new Deck(suit, value);`
 - `myDeck = Deck[NUMCARDS];`
 - `myDeck = new Card(suit, value);`
 - `myDeck = new Card[NUMCARDS];`

7.9. 8. Which of the following is correct <code to insert the heart cards into myDeck> so that the specification for the myDeck array is satisfied?

- (A) if ($i / 13 == 1$)
 myDeck[$i / 13$] = new Card("hearts", $i \% 13$);
- (B) if ($i \geq 13 \&\& i \leq 25$)
 myDeck[$i \% 13$] = new Card("hearts", $i \% 13$);
- (C) if ($i / 13 == 1$)
 myDeck[i] = new Card("hearts", $i \% 13$);
- (D) if ($i \geq 13 \&\& i \leq 25$)
 myDeck[i] = new Card("hearts", $i / 13$);
- (E) if ($i / 13 == 1$)
 myDeck[$i \% 13$] = new Card("hearts", $i \% 13$);

9. Consider the implementation of a writeDeck method that is added to the Deck class.

```
//Write the cards in myDeck, one per line
public void writeDeck()
{
    <implementation code>
}
```

Which of the following is correct <implementation code>?

- I System.out.println(myDeck);
 - II for (int i=0; i<NUMCARDS; i++)
 System.out.println(myDeck[i]);
 - III for (int i=0; i<NUMCARDS; i++)
 System.out.println(myDeck[i].toString());
- (A) I only
 - (B) II only
 - (C) III only
 - (D) I and III only
 - (E) II and III only

10. A method is to be written to search an array for a value that is larger than a given item and return its index. The problem specification does not indicate what should be returned if there are several such values in the array. Which of the following actions would be best?

- (A) The method should be written on the assumption that there is only one value in the array that is larger than the given item.
- (B) The method should be written so as to return the index of every occurrence of a larger value.
- (C) The specification should be modified to indicate what should be done if there is more than one index of larger values.
- (D) The method should be written to output a message if more than one larger value is found.
- (E) The method should be written to delete all subsequent larger items after a suitable index is returned.

11. When will method `whatIsIt` cause a stack overflow (i.e., cause computer memory to be exhausted)?

```
public static int whatIsIt(int x, int y)
{
    if (x > y)
        return x*y;
    else
        return whatIsIt(x-1, y);
}
```

- (A) Only when $x < y$
- (B) Only when $x \leq y$
- (C) Only when $x > y$
- (D) For all values of x and y
- (E) The method will never cause a stack overflow.

12. The boolean expression `a[i] == max || !(max != a[i])` can be simplified to

- (A) `a[i] == max`
- (B) `a[i] != max`
- (C) `a[i] < max || a[i] > max`
- (D) `true`
- (E) `false`

13. Suppose an `ArrayList` `list` is initialized with `Integer` values. Which of the following will cause an `IndexOutOfBoundsException` to be thrown?

- (A)

```
for (int i=0; i<=list.size(); i++)
    list.set(i, new Integer(0));
```
- (B) `list.add(list.size(), new Integer(0));`
- (C) `Integer int0b = list.get(list.size());`
- (D) `Integer int0b = list.remove(list.size());`
- (E) `list.add(-1, new Integer(0));`

Questions 14–19 refer to the Point, Quadrilateral, and Rectangle classes below:

```
public class Point
{
    private int xCoord;
    private int yCoord;

    //constructor
    public Point(int x, int y)
    {
        xCoord = x;
        yCoord = y;
    }

    //Return x coordinate of Point
    public int get_x()
    { return xCoord; }

    //Return y coordinate of Point
    public int get_y()
    { return yCoord; }

    //Change Point to new_x and new_y
    public void setPoint(int new_x, int new_y)
    {
        xCoord = new_x;
        yCoord = new_y;
    }
}

public abstract class Quadrilateral
{
    private String myLabels;      //e.g., "ABCD"

    //constructor
    public Quadrilateral(String labels)
    { myLabels = labels; }

    public String getLabels()
    { return myLabels; }

    public abstract int perimeter();
    public abstract int area();
}
```

ng will not
be exhausted)?

PAGE

GO ON TO THE NEXT PAGE

```

public class Rectangle extends Quadrilateral
{
    private Point myTopLeft; //coords of top left corner
    private Point myBotRight; //coords of bottom right corner

    //constructor
    public Rectangle(String labels, Point topLeft, Point botRight)
    { implementation }

    public int perimeter()
    { implementation }

    public int area()
    { implementation }

    /* Return true if Rectangle is valid, false otherwise.
     * Rectangle is valid if and only if "top" y-coord > "bottom" y-coord,
     * and "right" x-coord > "left" x-coord */
    public boolean isValid()
    { implementation }

    //Return top left corner Point
    public Point getTopLeft()
    { return myTopLeft; }

    //Return bottom right corner Point
    public Point getBotRight()
    { return myBotRight; }

    //Changes the top left and bottom right corner coordinates
    //of the current Rectangle
    public void alter(Point newTopLeft, Point newBotRight)
    { implementation }
}

```

14. Which statement about the Quadrilateral class is *false*?
- The perimeter and area methods are abstract because there's no suitable default code for them.
 - The getLabels method is not abstract because any subclasses of Quadrilateral will have the same code for this method.
 - If the Quadrilateral class is used in a program, it *must* be used as a superclass for at least one other class.
 - No instances of a Quadrilateral object can be created in a program.
 - Any subclasses of the Quadrilateral class *must* provide implementation code for the perimeter and area methods.

15. Which represents correct *implementation* code for the Rectangle constructor?

- I super(labels);
- II super(labels, topLeft, botRight);
- III super(labels);

myTopLeft = topLeft;

myBotRight = botRight;

- (A) I only
- (B) II only
- (C) III only
- (D) I and II only
- (E) II and III only

16. Which is a *true* statement about the Rectangle class?

- I The Rectangle class implements the Quadrilateral interface.
 - II The Rectangle class is a superclass of the Point class.
 - III The isValid method will allow clients of the class to check the validity of any given Rectangle object.
- (A) I only
 - (B) II only
 - (C) III only
 - (D) I and II only
 - (E) II and III only

17. A client program of the Rectangle class contains the following method:

```
//Precondition: rect is a valid rectangle
//Postcondition: returns true if rect is a square, false otherwise
public static boolean isSquare(Rectangle rect)
{
    return myBotRight.get_x() - myTopLeft.get_x() ==
           myTopLeft.get_y() - myBotRight.get_y();
}
```

This code is incorrect. Why?

- (A) The return statement is mathematically incorrect. It should be

```
return Math.abs(myBotRight.get_x() - myTopLeft.get_x()) ==
       Math.abs(myTopLeft.get_y() - myBotRight.get_y());
```

- (B) The return statement is syntactically incorrect. It should be

```
return myBotRight.x - myTopLeft.x ==
       myTopLeft.y - myBotRight.y;
```

- (C) The condition is insufficient: All four sides of the rectangle must be tested for equality.
- (D) The variables myTopLeft and myBotRight cannot be used in a client program.
- (E) The return type is incorrect. The method should return true or false.

GO ON TO THE NEXT PAGE

Consider the following classes for Questions 18 and 19:

```
public class Parallelogram extends Quadrilateral
{
    //private instance variables and constructor
    ...

    public int perimeter()
    { /* implementation not shown */ }

    public int area()
    { /* implementation not shown */ }
}

public class Square extends Rectangle
{
    //private instance variables and constructor
    ...

    public int perimeter()
    { /* implementation not shown */ }

    public int area()
    { /* implementation not shown */ }
}
```

18. A client program has this code segment:

```
Quadrilateral q1 = new Parallelogram(<parameter list>);
Quadrilateral q2 = new Square(<parameter list>);
System.out.println("Area of " + q1.getLabels() + " is " + q1.area());
System.out.println("Area of " + q2.getLabels() + " is " + q2.area());
```

Which is a *true* statement about this code?

- (A) The code will cause a compile-time error because there is no *getLabels* method in the *Parallelogram* and *Square* classes.
- (B) The code will cause a compile-time error because the *area* method in the *Quadrilateral* class is abstract.
- (C) The code will run as intended: *q1.area()* will give the area of the appropriate parallelogram, and *q2.area()* will give the area of the appropriate square.
- (D) The declarations are incorrect. They need to be changed as follows:
`Parallelogram q1 = new Parallelogram(<parameter list>);
Square q2 = new Square(<parameter list>);`

- (E) The output statements are incorrect. They need to be changed as follows:
`System.out.println("Area of " + q1.getLabels() + " is "
+ ((Parallelogram) q1).area());
System.out.println("Area of " + q2.getLabels() + " is "
+ ((Square) q2).area());`

20. R

19. Consider an `ArrayList`, `quadList`, of Quadrilateral objects: type `Rectangle`, `Parallelogram`, or `Square`. Refer to the following method, `writeAreas`:

```
/* Precondition: quadList contains Rectangle, Parallelogram, or
 * Square objects in an unspecified order
 * Postcondition: area of each Quadrilateral in quadList has been printed */
public static void writeAreas(ArrayList quadList)
{
    for (int i=0; i<quadList.size(); i++)
        < code to print area of Quadrilateral >
}
```

Which is correct <code to print area of Quadrilateral>?

- (A) `System.out.println("Area of " + quadList.getLabels()
+ " is " + quadList.area());`
- (B) `System.out.println("Area of " + quadList[i].getLabels()
+ " is " + quadList[i].area());`
- (C) `System.out.println("Area of " + (quadList.get(i)).getLabels()
+ " is " + (quadList.get(i)).area());`
- (D) `System.out.println("Area of " + ((Quadrilateral) quadList.get(i)).getLabels()
+ " is " + ((Quadrilateral) quadList.get(i)).area());`
- (E) `System.out.println("Area of " + ((Quadrilateral) quadList[i]).getLabels()
+ " is " + ((Quadrilateral) quadList[i]).area());`

20. Refer to the `doSomething` method:

```
//<postcondition>
public static void doSomething(ArrayList a, int i, int j)
{
    Object temp = a.get(i);
    a.set(i, a.get(j));
    a.set(j, temp);
}
```

Which best describes the <postcondition> for `doSomething`?

- (A) Removes from a the objects indexed at i and j.
- (B) Replaces in a the object indexed at i with the object indexed at j.
- (C) Replaces in a the object indexed at j with the object indexed at i.
- (D) Replaces in a the objects indexed at i and j with temp.
- (E) Interchanges in a the objects indexed at i and j.

Questions 21–23 refer to the `NegativeReal` class below, which defines a negative real number object.

```
public class NegativeReal
{
    private Double myNegReal;

    //constructor
    //Precondition: num < 0
    public NegativeReal(double num)
    { implementation }

    //Postcondition: returns value of this NegativeReal
    public double getValue()
    { implementation }

    //Postcondition: returns this NegativeReal rounded to the nearest integer
    public int getRounded()
    { implementation }
}
```

21. Which is a correct *implementation* for the constructor of a `NegativeReal` object?

- (A) `myNegReal = num;`
- (B) `myNegReal = Double(num);`
- (C) `myNegReal = new Double(num);`
- (D) `myNegReal = new Double(-num);`
- (E) `myNegReal = -(new Double(num));`

22. Which is a correct *implementation* of the `getValue` method?

- I `return myNegReal;`
- II `return myNegReal.doubleValue();`
- III `return doubleValue();`

- (A) I only
- (B) II only
- (C) III only
- (D) II and III only
- (E) I, II, and III

23. Here are some rounding examples:

Negative real number	Rounded to nearest integer
-3.5	-4
-8.97	-9
-5.0	-5
-2.487	-2
-0.2	0

Which implementation of getRounded produces the desired postcondition?

- (A) `return (int) (getValue() - 0.5);`
- (B) `return (int) (getValue() + 0.5);`
- (C) `return (int) getValue();`
- (D) `return myNegReal.intValue();`
- (E) `return getValue().intValue();`

24. Consider the following method.

```
public static void whatsIt(int n)
{
    if (n > 10)
        whatsIt(n/10);
    System.out.print(n % 10);
}
```

What will be output as a result of the method call `whatsIt(347)?`

- (A) 74
- (B) 47
- (C) 734
- (D) 743
- (E) 347

25. A large list of numbers is to be sorted into ascending order. Which of the following is a *true* statement?

- (A) If the array is initially sorted in descending order, then insertion sort will be more efficient than selection sort.
- (B) The number of comparisons for selection sort is independent of the initial arrangement of elements.
- (C) The number of comparisons for insertion sort is independent of the initial arrangement of elements.
- (D) The number of data movements in selection sort depends on the initial arrangement of elements.
- (E) The number of data movements in insertion sort is independent of the initial arrangement of elements.

26. Consider the code segment

```
if (n == 1)
    k++;
else if (n == 4)
    k += 4;
```

Suppose that the given segment is rewritten in the form

```
if (< condition >)
    < assignment statement >;
```

Given that n and k are integers and that the rewritten code performs the same task as the original code, which of the following could be used as

- (1) < condition > and (2) < assignment statement >?

- | | |
|--------------------------|---------------|
| (A) (1) n == 1 && n == 4 | (2) k += n |
| (B) (1) n == 1 && n == 4 | (2) k += 4 |
| (C) (1) n == 1 n == 4 | (2) k += 4 |
| (D) (1) n == 1 n == 4 | (2) k += n |
| (E) (1) n == 1 n == 4 | (2) k = n - k |

27. Which of the following will execute *without* throwing an exception?

I String s = null;
 String t = "";
 if (s.equals(t))
 System.out.println("empty strings?");

II String s = "holy";
 String t = "moly";
 if (s.equals(t))
 System.out.println("holy moly!");

III String s = "holy";
 String t = s.substring(4);
 System.out.println(s + t);

- (A) I only
 (B) II only
 (C) III only
 (D) I and II only
 (E) II and III only

28. Three numbers a , b , and c are said to be a *Pythagorean Triple* if and only if the sum of the squares of two of the numbers equals the square of the third. A programmer writes a method `isPythTriple` to test if its three parameters form a Pythagorean Triple:

```
//Returns true if a2 + b2 = c2; otherwise returns false
public static boolean isPythTriple(double a, double b, double c)
{
    double d = Math.sqrt(a*a + b*b);
    return d == c;
}
```

When the method was tested with known Pythagorean Triples, `isPythTriple` sometimes erroneously returned false. What was the most likely cause of the error?

- (A) Round-off error was caused by calculations with floating-point numbers.
- (B) Type `boolean` was not recognized by an obsolete version of Java.
- (C) An overflow error was caused by entering numbers that were too large.
- (D) c and d should have been cast to integers before testing for equality.
- (E) Bad test data were selected.

29. Refer to the following class, containing the `mystery` method.

```
public class SomeClass
{
    private int[] arr;

    //Constructor. Initializes arr to contain nonnegative
    //integers k such that 0 ≤ k ≤ 9
    public SomeClass()
    { /* implementation not shown */ }

    public int mystery()
    {
        int value = arr[0];
        for (int i=1; i < arr.length; i++)
            value = value*10 + arr[i];
        return value;
    }
}
```

Which best describes what the `mystery` method does?

- (A) It sums the elements of `arr`.
- (B) It sums the products $10*arr[0] + 10*arr[1] + \dots + 10*arr[arr.length-1]$.
- (C) It builds an integer of the form $d_1d_2d_3\dots d_n$, where $d_1 = arr[0]$, $d_2 = arr[1]$, ..., $d_n = arr[arr.length-1]$.
- (D) It builds an integer of the form $d_1d_2d_3\dots d_n$, where $d_1 = arr[arr.length-1]$, $d_2 = arr[arr.length-2]$, ..., $d_n = arr[0]$.
- (E) It converts the elements of `arr` to base 10.

Questions 30 and 31 refer to the search method in the Searcher class below.

```

public class Searcher
{
    private int[] arr;

    //Constructor. Initializes arr with integers
    public Searcher()
    { /* implementation not shown */ }

    /* Precondition: arr[first]...arr[last] sorted in ascending order
     * Postcondition: returns index of key in arr. If key not in arr,
     *                 returns -1 */
    public int search(int first, int last, int key)
    {
        int mid;
        while (first <= last)
        {
            mid = (first + last)/2;
            if (arr[mid] == key)           //found key, exit search
                return mid;
            else if (arr[mid] < key)      //key to right of arr[mid]
                first = mid + 1;
            else                          //key to left of arr[mid]
                last = mid - 1;
        }
        return -1;                      //key not in list
    }
}

```

30. Which assertion is true just before each execution of the while loop?

- (A) $\text{arr}[\text{first}] < \text{key} < \text{arr}[\text{last}]$
- (B) $\text{arr}[\text{first}] \leq \text{key} \leq \text{arr}[\text{last}]$
- (C) $\text{arr}[\text{first}] < \text{key} < \text{arr}[\text{last}]$ or key is not in arr
- (D) $\text{arr}[\text{first}] \leq \text{key} \leq \text{arr}[\text{last}]$ or key is not in arr
- (E) $\text{key} \leq \text{arr}[\text{first}]$ or $\text{key} \geq \text{arr}[\text{last}]$ or key is not in arr

31. Consider the array a with values as shown:

4, 7, 19, 25, 36, 37, 50, 100, 101, 205, 220, 271, 306, 321

where 4 is $a[0]$ and 321 is $a[13]$. Suppose that the search method is called with $\text{first} = 0$ and $\text{last} = 13$ to locate the key 205. How many iterations of the while loop must be made in order to locate it?

- (A) 3
- (B) 4
- (C) 5
- (D) 10
- (E) 13

GO ON TO THE NEXT PAGE

Refer to the RandomList class for Questions 32 and 33.

```
import java.util.*;  
  
public class RandomList  
{  
    private ArrayList myList;  
  
    //constructor  
    public RandomList()  
    { myList = getList(); }  
  
    /* Read random Integers from 0 to 100 inclusive into ArrayList list */  
    public ArrayList getList()  
    {  
        System.out.println("How many integers? ");  
        int listLength = IO.readInt(); //read user input  
        ArrayList list = new ArrayList();  
        Random r = new Random();  
        for (int i=0; i<listLength; i++)  
        {  
            < code to add Integer to list >  
        }  
        return list;  
    }  
  
    /* Print all elements of this list */  
    public void printList()  
    {  
        < code to print myList >  
    }  
}
```

32. Which represents correct < code to add Integer to list >?

- (A) list[i] = new Integer(r.nextInt(101));
- (B) list.add(new Integer(r.nextInt(101)));
- (C) list[i] = new Integer(r.nextInt(100));
- (D) list.add(new Integer(r.nextInt(100)));
- (E) int num = r.nextInt(101);
 list.add(new num);

520 PRACTICE EXAMS

33. Which represents correct <code to print myList>?

- I for (int i=0; i<myList.size(); i++)
 System.out.print(myList.get(i) + " ");
 - II for (int i=0; i<myList.size(); i++)
 System.out.print((Integer) myList.get(i) + " ");
 - III System.out.print(myList);
- (A) I only
(B) II only
(C) III only
(D) II and III only
(E) I, II, and III

Questions 34 and 35 refer to method `insert` described here. The `insert` method has two string parameters and one integer parameter. `insert` returns the string obtained by inserting the second string into the first starting at the position indicated by the integer parameter. For example, if `str1` contains `xy` and `str2` contains `cat`, then

<code>insert(str1, str2, 0)</code>	returns	<code>catxy</code>
<code>insert(str1, str2, 1)</code>	returns	<code>xcaty</code>
<code>insert(str1, str2, 2)</code>	returns	<code>xycat</code>

Here is the header for method `insert`.

```
//Precondition: 0 <= pos <= str1.length()
//Postcondition: returns <somestring>
public static String insert(String str1, String str2, int pos);
```

34. If $\text{str1} = a_0a_1\dots a_{n-1}$ and $\text{str2} = b_0b_1\dots b_{m-1}$, which of the following is a correct replacement for `<somestring>`?

- (A) $a_0a_1\dots a_{\text{pos}}b_0b_1\dots b_{m-1}a_{\text{pos}+1}a_{\text{pos}+2}\dots a_{n-1}$
- (B) $a_0a_1\dots a_{\text{pos}+1}b_0b_1\dots b_{m-1}a_{\text{pos}+2}a_{\text{pos}+3}\dots a_{n-1}$
- (C) $a_0a_1\dots a_{\text{pos}-1}b_0b_1\dots b_{m-1}a_{\text{pos}}a_{\text{pos}+1}\dots a_{n-1}$
- (D) $a_0a_1\dots a_{n-1}b_0b_1\dots b_{m-1}$
- (E) $a_0a_1\dots a_{\text{pos}-1}b_0b_1\dots b_{\text{pos}-1}a_{\text{pos}}a_{\text{pos}+1}\dots a_{n-1}$

35. Method `insert` follows:

```
//Postcondition: returns <somestring>
public static String insert(String str1, String str2, int pos)
{
    String first, last;
    <more code>
    return first + str2 + last;
}
```

Which of the following is a correct replacement for `<more code>`?

- (A) `first = str1.substring(0, pos);`
`last = str1.substring(pos);`
- (B) `first = str1.substring(0, pos-1);`
`last = str1.substring(pos);`
- (C) `first = str1.substring(0, pos+1);`
`last = str1.substring(pos+1);`
- (D) `first = str1.substring(0, pos);`
`last = str1.substring(pos+1, str1.length());`
- (E) `first = str1.substring(0, pos);`
`last = str1.substring(pos, str1.length()+1);`