Practice Test 1

AP® Computer Science A Exam

SECTION I: Multiple-Choice Questions

DO NOT OPEN THIS BOOKLET UNTIL YOU ARE TOLD TO DO SO.

At a Glance

Total Time
1 hour 30 minutes
Number of Questions
40
Percent of Total Score
50%
Writing Instrument
Pencil required

Instructions

Section I of this examination contains 40 multiple-choice questions. Fill in only the ovals for numbers 1 through 40 on your answer sheet.

Indicate all of your answers to the multiple-choice questions on the answer sheet. No credit will be given for anything written in this exam booklet, but you may use the booklet for notes or scratch work. After you have decided which of the suggested answers is best, completely fill in the corresponding oval on the answer sheet. Give only one answer to each question. If you change an answer, be sure that the previous mark is erased completely. Here is a sample question and answer.

Sample Question

Chicago is a
(A) state
(B) city
(C) country
(D) continent
(E) county

Sample Answer

Use your time effectively, working as quickly as you can without losing accuracy. Do not spend too much time on any one question. Go on to other questions and come back to the ones you have not answered if you have time. It is not expected that everyone will know the answers to all the multiple-choice questions.

About Guessing

Many candidates wonder whether or not to guess the answers to questions about which they are not certain. Multiple-choice scores are based on the number of questions answered correctly. Points are not deducted for incorrect answers, and no points are awarded for unanswered questions. Because points are not deducted for incorrect answers, you are encouraged to answer all multiple-choice questions. On any questions you do not know the answer to, you should eliminate as many choices as you can, and then select the best answer among the remaining choices.

GO ON TO THE NEXT PAGE.

Quick Reference

```
class java.lang.Object
• boolean equals(Object other)
• String toString()

class java.lang.Integer
• Integer(int value)
• int intValue()
• Integer.MIN_VALUE                    // minimum value represented by an int or Integer
• Integer.MAX_VALUE                    // maximum value represented by an int or Integer

class java.lang.Double
• Double(double value)
• double doubleValue()

class java.lang.String
• int length()
• String substring(int from, int to)   // returns the substring beginning at from
                                       // and ending at to-1
• String substring(int from)           // returns substring(from, length())
• int indexOf(String str)              // returns the index of the first occurrence of str;
                                       // returns -1 if not found
• int compareTo(String other)          // returns a value < 0 if this is less than other
                                       // returns a value = 0 if this is equal to other
                                       // returns a value > 0 if this is greater than other

class java.lang.Math
• static int abs(int x)
• static double abs(double x)
• static double pow(double base, double exponent)
• static double sqrt(double x)
• static double random()               // returns a double in the range [0.0, 1.0)

interface java.util.List<E>
• int size()
• boolean add(E obj)                   // appends obj to end of list; returns true
• void add(int index, E obj)           // inserts obj at position index (0 ≤ index ≤ size),
                                       // moving elements at position index and higher
                                       // to the right (adds 1 to their indices) and adjusts size
• E get(int index)
• E set(int index, E obj)              // replaces the element at position index with obj
                                       // returns the element formerly at the specified position
• E remove(int index)                  // removes element from position index, moving elements
                                       // at position index + 1 and higher to the left
                                       // (subtracts 1 from their indices) and adjusts size
                                       // returns the element formerly at the specified position

class java.util.ArrayList<E> implements java.util.List<E>
```

GO ON TO THE NEXT PAGE.

COMPUTER SCIENCE A

SECTION I

Time—1 hour and 30 minutes

Number of Questions—40

Percent of total exam grade—50%

Directions: Determine the answer to each of the following questions or incomplete statements, using the available space for any necessary scratchwork. Then decide which is the best of the choices given and fill in the corresponding oval on the answer sheet. No credit will be given for anything written in the examination booklet. Do not spend too much time on any one problem.

Notes:

- Assume that the classes listed in the Quick Reference have been imported where appropriate.
- Assume that declarations of variables and methods appear within the context of an enclosing class.
- Assume that method calls that are not prefixed with an object or class name and are not shown within a complete class definition appear within the context of an enclosing class.
- Unless otherwise noted in the question, assume that parameters in the method calls are not `null` and that methods are called only when their preconditions are satisfied.

MULTIPLE CHOICE QUESTIONS **USE THIS SPACE FOR SCRATCHWORK**

1. Consider the following methods.

```
public void trial()
{
   int a = 10;
      int b = 5;
      doublevalues(a,b);
      System.out.print(b);
   System.out.print(a);
}

public void doublevalues(int c, int d)
{
      c = c * 2;
      d = d * 2;
   System.out.print(c);
   System.out.print(d);
}
```

What is printed as the result of the call `trial()`?

(A) `2010`
(B) `2010105`
(C) `2010510`
(D) `20102010`
(E) `20101020`

GO ON TO THE NEXT PAGE.

2. Consider the following method.

```
/**
 * Precondition: a > b > 0
 */
public static int mystery(int a, int b)
{
   int d = 0;
   for (int c = a; c > b; c--)
   {
      d=d+c;
   }
   return d;
}
```

What is returned by the call `mystery(x,y)`?

(A) The sum of all the integers greater than `y` but less than or equal to `x`
(B) The sum of all the integers greater than or equal to `y` but less than or equal to `x`
(C) The sum of all the integers greater than `y` but less than `x`
(D) The sum of all the integers greater than `y` but less than or equal to `x`
(E) The sum of all the integers less than `y` but greater than or equal to `x`

3. Consider the following method.

```
public void mystery (int  n)
{
   int k;
   for (k = 0 ; k < n ; k++)
   {
      mystery(k);
      System.out.print (n) ;
   }
}
```

What value is returned by the call `mystery(3)` ?

(A) `0123`
(B) `00123`
(C) `0010012`
(D) `00100123`
(E) `001001200100123`

4. Consider an array of integers.

```
4     10     1     2     6     7     3     5
```

If selection sort is used to order the array from smallest to largest values, which of the following represents a possible state of the array at some point during the selection sort process?

(A) `1  4  10  2  3  6  7  5`
(B) `1  2  4  6  10  7  3  5`
(C) `1  2  3  10  6  7  4  5`
(D) `4  3  1  2  6  7  10  5`
(E) `5  3  7  6  2  1  10  4`

GO ON TO THE NEXT PAGE.

5. Consider the following code segment:

```
int  k;
int  A[];
A = new int [7];
for (k = 0; k < A.length; k++)
{
   A[k] = A.length - k;
}
for (k = 0; k < A.length - 1; k++)
{
   A[k+1] = A[k];
}
```

What values will A contain after the code segment is executed?

(A) 1 1 2 3 4 5 6
(B) 1 2 3 4 5 6 7
(C) 6 6 5 4 3 2 1
(D) 7 7 6 5 4 3 2
(E) 7 7 7 7 7 7 7

GO ON TO THE NEXT PAGE.

Questions 6–7 refer to the following two classes.

```
public class PostOffice
{
    // constructor initializes boxes
    // to length 100
    public PostOffice( )
    {   /* implementation not shown   */}

    // returns the given p.o. box
    //  0 <= theBox  <  getNumBoxes ()
    public Box getBox (int theBox)
    {   /* implementation not shown   */}
    // returns the number of p.o. boxes
    public int getNumBoxes ()
    {   /* implementation not shown   */}

    // private data members and
    // other methods not shown
}

public class Box
{
    // constructor
    public Box ( )
    {   /* implementation not shown  */}

    // returns the number of this box
    public int getBoxNumber ( )
    {   /* implementation not shown   */}

    // returns the number of pieces
    // of mail in this box
    public int getMailCount ( )
    {   /* implementation not shown  */}
    // returns the given piece of mail
    // 0 <= thePiece < getMailCount ( )
    public Mail getMail (int thePiece)
    {   /* implementation not shown  */}
    // true if the box has been assigned
    // to a customer
    public boolean isAssigned ( )
    {   /* implementation not shown  */}
    // true if the box contains mail
    public boolean hasMail ( )
    {   /* implementation not shown  */}
    // private data members and
    // other methods not shown
}
public class Mail
{
    // private members,  constructors, and
    // other methods not shown
}
```

GO ON TO THE NEXT PAGE.

6. Consider the following code segment:

```
PostOffice p[ ]
p = new PostOffice[10];
```

Assuming that the box has been assigned and that it has at least four pieces of mail waiting in it, what is the correct way of getting the fourth piece of mail from the 57th box of the tenth post office of `p`?

(A) `Mail m = p[10].getBox(57).getmail(4);`
(B) `Mail m = p[9].getBox(56).getMail (3);`
(C) `Mail m = p.getMail(57).getMail (4) [10];`
(D) `Mail m = getMail(getBox(p[9], 560, 3);`
(E) `Mail m = new Mail(10, 57, 4);`

GO ON TO THE NEXT PAGE.

7. Consider the incomplete function `printEmptyBoxes` given below. `printEmptyBoxes` should print the box numbers of all of the boxes that do not contain mail.

```
public void printEmptyBoxes (PostOffice p [ ])
{
   for (int k = 0; k < p.length - 1 ; k++)
   {
      for (int x = 0; x < p[k].getNumBoxes( ) - 1 ; x++)
      {
           //  missing expression
      }
   }
}
```

Which of the following could be used to replace

```
// missing expression
```

body so that `printBoxesWithoutMail` works as intended?

(A)
```
if (p[k].getBox(x).isAssigned( ) &&
!p[k].getBox(x).hasMail( ) )
{
     System.out.println(P[k].getBox(x).getBoxNumber( )) ;
}
```

(B)
```
if (p[x].getBox(k).isAssigned ( ) &&
!p[x].getBox(k).hasMail( ))
{
 System.out.println(p[x].getBox(k).getBoxNumber( ));
}
```

(C)
```
if (p[k].getBox(x).isAssigned( ) &&
!p[k].getBox(x).hasMail( ))
{
     System.out.println (p[k].getBoxNumber (x));
}
```

(D)
```
if (p[x].getBox(k).isAssigned( ) &&
!p[x].getBox (k).hasMail( ))
{
     System.out.println(p[x].getBoxNumber(k));
 }
```

(E)
```
if (p[x].getBox(k).isAssigned( ) &&
p[x].getBox(k).getMail( )   ==   0)
{
     System.out.println(k);
}
```

GO ON TO THE NEXT PAGE.

8. Assume that a and b are Boolean variables that have been initialized. Consider the following code segment.

```
a = a && b;
b = a | |  b;
```

Which of the following statements is true?

I. The final value of a is the same as the initial value of a.
II. The final value of b is the same as the initial value of b.
III. The final value of a is the same as the initial value of b.

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

9. Consider the following code segment.

```
int x;
x = 53;
if (x > 10)
{
   System.out.print("A") ;
}
if (x > 30)
{
   System.out.print("B") ;
}
else if (x > 40)
{
   System.out.print("C") ;
}
if (x > 50)
{
   System.out.print ("D") ;
}
if (x > 70)
{
   System.out.print ("E") ;
}
```

What is output when the code is executed?

(A) A
(B) D
(C) ABD
(D) ABCD
(E) ABCDE

GO ON TO THE NEXT PAGE.

10. Consider the following code segment:

```
int j;
int k;
for (j = -2; j <= 2; j = j + 2)
{
   for (k = j; k < j + 3; k++)
   {
      System.out.print(k + " " ) ;
   }
}
```

What is the output when the code is executed?

(A) -2 -1 0
(B) -2 -1 0 1 2
(C) 0 1 2 0 1 2 0 1 2
(D) -2 0 2
(E) -2 -1 0 0 1 2 2 3 4

11. Consider the following method.

```
public void mystery (int  count, String s)
{
   if  (count  <=  0)
   {
      return;
   }
   if  (count % 3 == 0)
   {
      System.out.print(s + "--" + s)
   }
   else if (count %  3 == 1)
   {
      System.out.print(s + "-" + s)
   }
   else
   {
      System.out.print (s) ;
   }
   mystery(count - 1, s) ;
}
```

What is output by the call `mystery(5, "X");`?

(A) XX - XX - - XXX - X
(B) XX - XX - XX - XX
(C) XXX - - XX - X - XX - - XXX
(D) XX - XXX - - XXX - XX
(E) XXXXX

GO ON TO THE NEXT PAGE.

Questions 12–13 refer to the following classes and method descriptions.

Class `Table` has a method, `getPrice`, which takes no parameters and returns the price of the table.

Class `Chair` also has a method, `getPrice`, which takes no parameters and returns the price of the chair.

Class `DiningRoomSet` has a constructor which is passed a `Table` object and an `ArrayList` of `Chair` objects. It stores these parameters in its private data fields `myTable` and `myChairs`.

Class `DiningRoomSet` has a method, `getPrice`,which takes no parameters and returns the price of the dining room set. The price of a dining room set is calculated as the sum of the price of its table and all of its chairs.

12. What is the correct way to define the signature of the constructor for the `DiningRoomSet` class?

(A) `public void DiningRoomSet (Table t, ArrayList, chairs)`
(B) `public DiningRoomSet (Table t, ArrayList chairs)`
(C) `public void DiningRoomSet (Table t, ArrayList Chair Chairs)`
(D) `public DiningRoomSet (Table t, ArrayList Chair Chairs)`
(E) `public DiningRoomSet (Table t, Chair Chairs)`

13. What is the correct way to implement the getPrice method of the `DiningRoomSet` class?

(A)
```
public double getPrice(Table t, ArrayList chairs)
{
     return t.getPrice() + chairs.getPrice() ;
}
```
(B)
```
public double getPrice(Table t, ArrayList chairs)
{
     return myTable.getPrice() + myChairs.getPrice();
}
```
(C)
```
public double getPrice()
{
     return myTable.getPrice() + myChairs.getPrice();
}
```
(D)
```
public double getPrice()
{
     double result  = myTable.getPrice();
     for (int k = 0; k < myChairs.size() - 1; k++)
     {
           result += ((Chair)myChairs.get(k)).getPrice();
     }
     return result;
}
```
(E)
```
public double getPrice()
{
      double result = myTable.getPrice() ;
      for (int k = 0; k  <  myChairs.length - 1; k++)
      {
           result += ((Chair)myChairs[k]).getPrice( );
      }
      return result ;
}
```

GO ON TO THE NEXT PAGE.

14. Consider the following output:

```
6   5   4   3   2   1
5   4   3   2   1
4   3   2   1
3   2   1
2   1
1
```

Which of the following code segments produces the above output when executed?

(A)
```
for  (int j = 6;  j  <  0;  j--)
{
  for (int k = j; k > 0 ; k --)
  {
      System.out.print (k  +  "   ") ;
  }
  System.out.println("") ;
}
```

(B)
```
for (int j = 6; j >= 0; j--)
{
  for (int k = j; k >= 0; k--)
  {
      System.out.print(k + " ");
  }
  System.out.println (" ") ;
}
```

(C)
```
for (int j = 0; j < 6; j++)
{
  for (int k = 6 - j; k > 0; k--)
  {
      System.out.print (k + "  ") ;
  }
  system.out.println (" ") ;
}
```

(D)
```
for (int j = 0; j < 6; j++)
{
  for (int k = 7 - j ; k > 0 ; k --)
  {
      System.out.print(k + " ");
  }
  System.out.println (" ") ;
}
```

(E)
```
for (int j = 0; j < 6; j++)
{
  for (int k = 6  -  j ; k >= 0; k--)
  {
      System.out.print(k + " ");
  }
  System.out.println (" ") ;
}
```

GO ON TO THE NEXT PAGE.

15. Consider the following code segment.

```
List<Integer> list = new ArrayList<Integer>();
list.add(new Integer (7));
list.add (new Integer (6));
list.add (1, new Integer (5));
list.add (1, new Integer (4));
list.add (new Integer (3));
list.set (2, new Integer (2));
list.add (1, new Integer (1));
System.out.prinln (list);
```

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

(A) `[1, 4, 2, 7, 6, 3]`
(B) `[7, 1, 4, 2, 6, 3]`
(C) `[7, 2, 5, 4, 3, 1]`
(D) `[7, 6, 2, 4, 3, 1]`
(E) `[7, 1, 2]`

16. Consider the following declarations.

```
public interface Animal
{
    String makeSound();
    String animalType();
}

    public static class Dog implements Animal
{
    public String makeSound(Animal a)
    {
       // Implementation not shown
    }
}
```

Which of the following methods must be included in the declaration of the `Dog` class in order for the class to successfully compile?

I. `public String makeSound()`
II. `public String animalType()`
III. `public String animalType(animal b)`

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

GO ON TO THE NEXT PAGE.

17. Consider the following two classes.

```
public static class Fish
{
  public String endoskeleton = "bone";

  public void action()
  {
      System.out.println("splash splash");
  }
}

public static class Shark extends Fish
{
  public void action()
  {
      System.out.println("chomp chomp");
  }

  public String endoskeleton="cartilage";
}
```

Which of the following is the correct output after the following code segment is executed?

```
Fish Bob = new Shark();
System.out.println(Bob.endoskeleton);
Bob.action();
```

(A) bone
chomp chomp

(B) bone
splash splash

(C) cartilage
splash splash

(D) cartilage
chomp chomp

(E) cartilage
splash splash
chomp chomp

GO ON TO THE NEXT PAGE.

Questions 18–19 refer to the following incomplete method.

18. The following insertSort method sorts the values in an integer array, sort, in ascending order.

```
Line 1:   public static void insertSort(int[] sort)
Line 2:         {
Line 3:               for (int index=1;index<sort.length;index++)
Line 4:               {
Line 5:         int temp=sort[index];
Line 6:         while (index > 0 && sort[index-1]>temp)
Line 7:                     {
Line 8:               // Missing code
Line 9:                     }
Line 10:        sort[index]=temp;
Line 11:              }
Line 12:        }
```

Which of the following can be used to replace "`// Missing code`" so that the `insertSort` method will execute properly?

(A)
```
sort[index]=sort[index-1];
index++;
```
(B)
```
sort[index-1]=sort[index];
index--;
```
(C)
```
sort[index]=sort[index+1];
index++;
```
(D)
```
sort[index]=sort[index-1];
index--;
```
(E)
```
sort[index]=sort[index+1];
index--;
```

19. Assuming that the "`// Missing code`" is implemented properly, what change can be made to the code in order for the array to be sorted in descending order?

(A) Replace Line 6 with: `while (index < 0 && sort[index-1]<temp)`
(B) Replace Line 6 with: `while (index < 0 && sort[index-1]<temp)`
(C) Replace Line 6 with: `while (index > 0 && sort[index-1]<temp)`
(D) Replace Line 3 with: `for (int index=sort.length-1;index>0;index--)`
(E) Replace Line 3 with: `for (int index=1;index>0;index--)`

20. Which of the following arrays would be sorted the slowest using insertion sort?

(A) `[3 4 6 2 7 3 9]`
(B) `[3 2 5 4 6 7 9]`
(C) `[9 7 6 5 4 3 2]`
(D) `[2 3 4 5 6 7 9]`
(E) `[9 3 2 4 5 7 6]`

GO ON TO THE NEXT PAGE.

Questions 21–23 refer to the following incomplete class declaration used to represent fractions with integral numerators and denominators.

```
public class Fraction
{
  private int numerator ;
  private int denominator ;

  public Fraction ( )
  {
      numerator = 0 ;
      denominator = 1 ;
  }

  public Fraction (int n, int d)
  {
      numerator = n;
      denominator = d;
  }

  // postcondition: returns the
  //  numerator
  public int getNumerator ( )
  {   /* implementation not shown */  }

  // postcondition: returns the
  //  denominator
  public int getDenominator ( )
  {   /* implementation not shown*/   }

  // postcondition: returns the greatest
  // common divisor of x and y
  public int gcd (int x, int y)
  {   /* implementation not shown*/   }

  // postcondition: returns the Fraction
  //  that is the result of multiplying
  //  this Fraction and f

  public Fraction multiply (Fraction f)
  {   /* implementation not shown */  }
  // . . . other methods not shown
}
```

GO ON TO THE NEXT PAGE.

21. Consider the method multiply of the Fraction class.

```
// postcondition: returns the Fraction
//     that is the result of multiplying
//     this Fraction and f
public Fraction multiply (Fraction f)
{  /* missing code*/    }
```

Which of the following statements can be used to replace `/* missing code */` so that the muliply method is corretly implemented?

I.
```
return Fraction (
  numerator * f.getNumerator ( ) ,
  denominator * f.getDenominator ( ) );
```

II.
```
return new Fraction (
  numerator * f.numerator ,
  denominator * f.denominator ( ) );
```

III.
```
return new Fraction (
  numerator * f.getNumerator ( ) ,
  denominator * f.getDenominator ( ) );
```

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

22. Consider the use of the Fraction class to muliply the fractions $\frac{3}{4}$ and $\frac{7}{19}$. Consider the following code:

```
Fraction fractionOne;
Fraction fractionTwo;
Fraction answer;
fractionOne = new Fraction (3, 4) ;
fractionTwo = new Fraction (7, 19) ;
/* missing code */
```

Which of the following could be used to replace `/* missing code */` so that answer contains the result of muliplying fractionOne by fractionTwo?

(A)
```
answer = fractionOne * fractionTwo
```
(B)
```
answer = multiply (fractionOne,fractionTwo) ;
```
(C)
```
answer = fractionOne.multiply (fractionTwo) ;
```
(D)
```
answer = new Fraction (fractionOne,fractionTwo) ;
```
(E)
```
answer = (fractionOne .getNumerator ( ) * fractionTwo .getNumerator ( ) ) /
      (fractionOne .getDenominator ( ) * fractionTwo .getDenominator ( ) ) ;
```

GO ON TO THE NEXT PAGE.

23. The following incomplete class declaration is intended to extend the `Fraction` class so that fractions can be manipulated in reduced form (lowest terms).

Note that a fraction can be reduced to lowest terms by dividing both the numerator and denominator by the greatest common divisor (gcd) of the numerator and denominator.

```
public class ReducedFraction extends Fraction
{
   private int reducedNumerator ;
   private int reducedDenominator ;
   // . . . constructors and other methods not shown
}
```

Consider the following proposed constructors for the `ReducedFraction` class:

I.
```
public ReducedFraction ( )
{
  reducedNumerator = 0;
  reducedDenominator = 1;
}
```

II.
```
public reducedFraction (int n, int d)
{
  numerator = n;
  denominator = d;
  reducedNumerator = n / gcd (n, d);
  reducedDenominator = d / gcd (n, d);
}
```

III.
```
public ReducedFraction (int n, int d)
{
  super (n, d) ;
  reducedNumerator = n / gcd (n, d) ;
  reducedDenominator = d / gcd (n, d) ;
}
```

Which of these constructor(s) would be legal for the `ReducedFraction` class?

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

GO ON TO THE NEXT PAGE.

24. Consider `s1` and `s2` defined as follows.

```
String s1 = new String("hello") ;
String s2 = new String("hello") ;
```

Which of the following is/are correct ways to see if `s1` and `s2` hold identical strings?

I.
```
if (s1 == s2)
  /* s1 and s2 are identical */
```
II.
```
if (s1 .compareTo (s2) == 0 )
  /* s1 and s2 are identical */
```
III.
```
if (s1 .equals (s2) )
  /* s1 and s2 are identical */
```

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

25. Consider the following variable and method declarations:

```
String s ;
String t ;
public void mystery (String a, String b)
{
   a = a + b ;
   b = b + a ;
}
```

Assume that `s` has the value "`Elizabeth`" and `t` has the value "`Andrew`" and mystery (`s`, `t`) is called. What are the values of `s` and `t` after the call to mystery?

	a	b
(A)	Elizabeth	Andrew
(B)	ElizabethAndrew	AndrewElizabeth
(C)	ElizabethAndrew	AndrewElizabethAndrew
(D)	ElizabethAndrew	ElizabethAndrewAndrew
(E)	ElizabethAndrewElizabeth	AndrewElizabethAndrew

GO ON TO THE NEXT PAGE.

26. Consider the following incomplete and *incorrect* class declaration:

```
public class Point implements Comparable
{
   private int x;
   private int y;
   public boolean compareTo (Point other)
   {
      return (x == other.x &&
              y == other.y);
   }
   // . . . constructors and other methods
   // not shown
}
```

For which of the following reasons is the above class declaration incorrect?

I. Objects may not access private data fields of other objects in the same class.
II. The `Comparable` interface requires that `compareTo` be passed an `Object` rather than a `Point`.
III. The `Comparable` interface requires that `compareTo` return an `int` rather than a `boolean`.

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

GO ON TO THE NEXT PAGE.

27. Consider the following abstraction of a `for` loop where `<1>, <2>, <3>,` and `<4>` represent legal code in the indicated locations:

```
for (<1>; <2>; <3>)
{
   <4>
}
```

Which of the following `while` loops has the same functionality as the above `for` loop?

(A)
```
<1> ;
while (<2>)
{
   <3>;
   <4>
}
```

(B)
```
<1> ;
while ( <2>)
{
   <4>
   <3> ;
}
```

(C)
```
<1> ;
   while ( !<2>)
{
   <3> ;
   <4>
}
```

(D)
```
<1> ;
while ( !<2>)
{
   <4>
   <3>;
}
```

(E)
```
<1> ;
<3> ;
while ( <2>)
{
   <4>
   <3> ;
}
```

28. Consider the following expression:

```
a / b + c - d % e * f
```

Which of the expressions given below is equivalent to the one given above?

(A) `( (a / b) + (c - d) ) % (e *f)`
(B) `( ( ( ( a / b) + c) - d) % e) * f`
(C) `( ( a / b) + c) - (d % (e * f)`
(D) `(a / ( (b + c) - d)  % e) * f`
(E) `( (a / b) + c) - ( (d % e) * f)`

GO ON TO THE NEXT PAGE.

29. Assume that a program declares and initializes x as follows:

```
String [] x ;
x = new String[10] ;
initialize(x);          // Fills the array x with
                        // valid strings each of
                        // length 5
```

Which of the following code segments correctly traverses the array and prints out the first character of all ten strings followed by the second character of all ten strings, and so on?

I.
```
int i;
int j;
for (i = 0 ; i < 10 ; i++)
     for (j = 0 ; j < 5 ; j++)
         System.out.print (x[i].substring (j, j+1));
```

II.
```
int i;
int j;
for (i = 0 ; i < 5 ; i++)
   for (j = 0 ; j < 10 ; j++)
       System.out.print (x[j].substring (i, i+1));
```

III.
```
int i ;
int j ;
for (i = 0 ; i < 5 ; i++)
   for (j = 0 ; j < 10 ; j++)
       System.out.print (x[i].substring (j, j+1));
```

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

30. Consider the following declaration and assignment statements:

```
int a = 7 ;
int b = 4 ;
double c ;
c = a / b ;
```

After the assignment statement is executed, what's the value of `c`?

(A) 1.0
(B) 1.75
(C) 2.0
(D) An error occurs because `c` was not initialized.
(E) An error occurs because because `a` and `b` are integers and `c` is a double.

GO ON TO THE NEXT PAGE.

31. Consider the following code segment:

```
int x ;
x = /* initialized to an integer */
if (x % 2 == 0 && x / 3 == 1)
   System.out.print("Yes") ;
```

For what values of x will the word "Yes" be printed when the code segment is executed?

(A) 0
(B) 4
(C) Whenever x is even and x is not divisible by 3
(D) Whenever x is odd and x is divisible by 3
(E) Whenever x is even and x is divisible by 3

32. Consider the following incomplete class definition:

```
public class SomeClass
{
   private String myName ;
   // postcondition: returns myName
   public String getName ( )
   { /* implmentation not shown */ }
   // postcondition: myName == name
   public void setName (String name)
   { /* implmentation not shown */ }
   // . . . constructors, other methods
   // and private data not shown
}
```

Now consider the method swap, not part of the SomeClass class.

```
// precondition: x and y are correctly
// constructed
// postcondition: the names of objects
// x and y are swapped
public void swap (SomeClass x, SomeClass y)
{
   <missing code>
}
```

GO ON TO THE NEXT PAGE.

Which of the following code segments can replace <missing code> so that the method swap works as intended?

I.
```
SomeClass temp ;
temp = x;
x = y ;
y = temp ;
```

II.
```
String temp ;
temp = x.myName ;
x .myName = y .myName
y .myName = temp ;
```

III.
```
String temp ;
temp = x .getName ( ) ;
x .setName (y .getName ( ) ) ;
y .setName (temp) ;
```

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

33. A bookstore wants to store information about the different types of books it sells.

For each book, it wants to keep track of the title of the book, the author of the book, and whether the book is a work of fiction or nonfiction.

If the book is a work of fiction, then the bookstore wants to keep track of whether it is a romance novel, a mystery novel, or science fiction.

If the book is a work of nonfiction, then the bookstore wants to keep track of whether it is a biography, a cookbook, or a self-help book.

Which of the following is the best design?

(A) Use one class, `Book`, which has three data fields: `String title`, `String author`, and `int bookType`.
(B) Use four unrelated classes: Book, Title, Author, and BookType.
(C) Use a class `Book` which has two data fields: `String title`, `String author`, and a subclass: `BookType`.
(D) Use a class `Book` which has two data fields: `String title`, `String author`, and two subclasses: `RomanceNovel`, `Mystery`, `ScienceFiction`, `Biography`, `Cookbook`, and `SelfHelpBook`.
(E) Use a class `Book` which has two data fields: `String title`, `String author`, and two subclasses: `FictionWork` and `NonFictionWork`. The class `FictionWork` has three subclasses, `RomanceNovel`, `Mystery`, and `ScienceFiction`. The class `NonFictionWork` has three subclasses: `Biography`, `Cookbook`, and `SelfHelpBook`.

GO ON TO THE NEXT PAGE.

34. Consider the following code:

```
public int mystery (int x)
{
   if (x == 1)
      return <missing value> ;
   else
      return (2 * mystery (x-1) ) + x ;
}
```

Which of the following can be used to replace `<missing value>` so that mystery (4) returns 34?

(A) 0
(B) 1
(C) 2
(D) 3
(E) 4

35. Consider the following code segment:

```
int [ ] X;
int [ ] Y;
X = initializeX ( ) ;    // returns a valid
                         // initialized int [ ]
Y = initializeY ( ) ;    // returns a valid
                         // initialized int [ ]
for (int k = 0 ;
   k < X.length && X[k] = = Y[k];
   k++)
{
   /* some code */
}
```

Assuming that after `X` and `Y` are initialized, `X.length == Y.length`, which of the following must be true after executing this code segment?

(A) `k < X.length`
(B) `k < X.length && X[k] == Y[k]`
(C) `k < X.length && X[k] ! = Y[k]`
(D) `k >= X.length | | X[k] == Y[k]`
(E) `k >= X.length | | X[k] ! = Y[k]`

36. Which of the following would *not* cause a run-time exception?

(A) Dividing an integer by zero
(B) Using an object that has been declared but not instantiated
(C) Accessing an array element with an array index that is equal to the length of the array
(D) Attempting to cast an object to a subclass of which it is not an instance
(E) Attempting to call a method with the wrong number of arguments

GO ON TO THE NEXT PAGE.

37. Assume that a and b are properly initialized variables of type `Double`.

Which of the following is an equivalent expression to:

```
a.doubleValue ( ) ! = b.doubleValue ( )
```

(A) `a ! = b`
(B) `a.notEquals (b)`
(C) `! (a.doubleValue ( ) .equals(b.doubleValue ( ) ) )`
(D) `! (a.compareTo(b) )`
(E) `a.compareTo(b) ! = 0`

38. Which of the following would be the least effective way of ensuring reliability in a program?

(A) Encapsulating functionality in a class by declaring all data fields to be public
(B) Defining and following preconditions and postconditions for every method
(C) Including assertions at key places in the code
(D) Using descriptive variable names
(E) Indenting code in a consistent and logical manner

39. Consider a dictionary that has 1,000 pages with 50 words on each page.

In order to look up a given target word, a student is considering using one of the following three methods:

Method 1

Use a binary search technique to find the correct page (comparing the target word with the first word on a given page). When the correct page is found, use a sequential search technique to find the target word on the page.

Method 2

Use a sequential search technique to find the correct page (comparing the target word with the first word on a given page). When the correct page is found, use another sequential search technique to find the target word on the page.

Method 3

Use a sequential search technique on all of the words in the dictionary to find the target word.

Which of the following best characterizes the greatest number of words that will be examined using each method?

	Method 1	Method 2	Method 3
(A)	10	50	1,000
(B)	55	500	2,500
(C)	55	525	25,000
(D)	60	1,050	1,050
(E)	60	1,050	50,000

GO ON TO THE NEXT PAGE.

40. Which of the following is *not* a peripheral?

(A) A color laser printer
(B) A monitor
(C) A word processing application
(D) A mouse
(E) An external CD-ROM drive

END OF SECTION I

**IF YOU FINISH BEFORE TIME IS CALLED,
YOU MAY CHECK YOUR WORK ON THIS SECTION.**

DO NOT GO ON TO SECTION II UNTIL YOU ARE TOLD TO DO SO.

COMPUTER SCIENCE A

SECTION II

Time—1 hour and 30 minutes

Number of Questions—4

Percent of Total Grade—50%

Directions: SHOW ALL YOUR WORK. REMEMBER THAT PROGRAM SEGMENTS ARE TO BE WRITTEN IN JAVA™.

Notes:

- Assume that the classes listed in the Java Quick Reference have been imported where appropriate.
- Unless otherwise noted in the question, assume that parameters in method calls are not null and that methods are called only when their preconditions are satisfied.
- In writing solutions for each question, you may use any of the accessible methods that are listed in classes defined in that question. Writing significant amounts of code that can be replaced by a call to one of these methods will not receive full credit.

FREE RESPONSE QUESTIONS

1. In a certain school, students are permitted to enroll in one elective class from a list of electives offered. Because there are a limited number of spaces in each class for students, and because some electives are more popular than others, a lottery system was devised by the school to assign students to electives.

   Each student lists three choices for electives. The school orders the students randomly and assigns each student to the first available elective in the student's list of three choices. If none of the three choices is available (because those electives are fully enrolled), the school does not assign the student to an elective.

   After the school attempts to assign all of the students to electives, it produces a list of students it was unable to assign.

   For example, suppose there are six electives available to students: Astronomy, Ballroom Dance, Basketweaving, Constitutional Law, Marine Biology, and Programming.

   The following table shows the name, maximum enrollment, and current enrollment for six electives after 64 students have been successfully assigned to electives:

Elective Name	Maximum Enrollment	Current Enrollment
Astronomy	12	12
Ballroom Dance	20	3
Basketweaving	15	14
Constitutional Law	10	7
Marine Biology	10	10
Programming	30	30

GO ON TO THE NEXT PAGE.

Note that three elctives, Astronomy, Programming, and Marine Biology, are fully enrolled and are no longer options for students.

Now suppose that the following students need to be assigned to electives:

Student	First Choice `getChoice (0)`	Second Choice `getChoice (1)`	Third Choice `getChoice (2)`
Andrew	Programming	Marine Biology	Ballroom Dance
David	Constitutional Law	Basketweaving	Programming
Elizabeth	Marine Biology	Programming	Astronomy
Ethan	Basketweaving	Marine Biology	Astronomy
Katharine	Programming	Basketweaving	Marine Biology

Andrew's first and second choices are fully enrolled, but his third choice has openings. Andrew will be enrolled in Ballroom Dance.

David's first choice has openings. David will be enrolled in Constitutional Law.

All three of Elizabeth's choices are fully enrolled. Elizabeth will remain unassigned to an elective.

Ethan's first choice has one opening left. Ethan will be enrolled in Basketweaving. Note that Basketweaving is now fully enrolled.

All three of Katharine's choices are now fully enrolled. Katharine will remain unassigned to an elective.

In this problem, the school is modeled by the class `School`. Students and electives are modeled by the classes `Student` and `Elective` respectively.

The `School` class includes the following methods and private data:

- `studentList`—This `ArrayList` holds the list of students in the order in which the students should be scheduled.
- `electiveList`—This `ArrayList` holds the electives that students may choose.
- `getElectiveByName`—This method returns the `Elective` in `electiveList` with the given name.
- `assignElectivesToStudents`—This method encapsulates the functionality of assigning students (if possible) their first, second, or third elective choice.
- `studentsWithoutElectives`—This method returns an `ArrayList` containing students that have not been assigned an elective.

GO ON TO THE NEXT PAGE.

```
public class School
{
    private ArrayList<Student> studentList;
    // each entry is an instance of a
    // Student representing one student
    // at the school; students are in
    // the order they should be scheduled

    private ArrayList<Elective> electiveList;
    // each entry is an instance of an
    // Elective representing one elective
    // offered at the school

    // precondition: name is the name of an
    //     Elective in electiveList
    // postcondition: returns the Elective
    //     in electiveList with the given
    //     name
    private Elective getElectiveByName (String name)
    { /* to be implemented in part (a) */ }
       // postcondition: returns the size
    // of electiveList
    private int getElectiveListSize()
{
       return electiveList.size();
    }
    private int getStudentListSize()
{
       return studentList.size();
    }

    // postcondition: All Students in
    //     studentList have been either
    //     assigned their first available
    //     elective choice or not assigned;
    //     All Electives in electiveList have
    //     been updated appropriately as
    //     Students are assigned to them
    public void assignElectivesToStudents ( )
    { /* to be implemented in part (b) */ }

    // postcondition: returns a list of
    //     those Students who have not been
    //     assigned an Elective
    public ArrayList<Student>
       studentsWithoutElectives ( )
    { /* to be implemented in part (c) */}

    // ... constructors, other methods,
    //     and other private data not shown
}
```

GO ON TO THE NEXT PAGE.

The Student class includes the following methods and private data:

- `getChoice`—This method returns the name of the given elective choice of the student. The first elective choice has index 0, the second has index 1, and the third has index 2.
- `hasElective`—This method returns true if the student has been assigned an elective; it returns false otherwise.
- `assignElective`—This method assigns the given elective to this student.

```
public class Student
{
    // precondition: 0 <= index < 3
    // postcondition: returns the name
    //    of the given elective choice
    public String getChoice (int index)
    { /* code not shown */}

    // postcondition: returns true if
    //    an Elective has been assigned
    //    to this Student
    public boolean hasElective ( )
    { /* code not shown */ }

    // precondition: e is not null
    // postcondition: e has been assigned
    //    to this Student; e has not been
    //    modified
    public void assignElective (Elective e)
    { /* code not shown */ }

    // ... constructors, other methods,
    // and other private data not shown
}
```

The Elective class includes the following methods:

- `getName`—This method returns the name of this elective.
- `getMaxClassSize`—This method returns the maximum number of students that can be assigned to this elective.
- `getClassSize`—This method returns the number of students that have been assigned to this elective.
- `addStudent`—This method assigns the given student to this elective.

GO ON TO THE NEXT PAGE.

```
public class Elective
{
    // postcondition: returns the name
    //    of this Elective
    public String getName ( )
    { /* code not shown */ }

    // postcondition: returns the
    //    maximum number of Students
    //    that can be added to this
    //    Elective
    public int getMaxClassSize ( )
    { /* code not shown */ }

    // postcondition: returns the
    //    number of Students that have
    //    been added to this Elective;
    //    0 < = getClassSize ( ) < =
    //    getMaxClassSize ( )
    public int getClassSize ( )
    { /* code not shown */ }

    // precondition: getClassSize ( ) <
    //    getMaxClassSize ( ); s is not null
    // postcondition: s has been added to
    //    this Elective; getClassSize ( ) has
    //    been increased by 1
    public void addStudent (Student s)
    { /* code not shown */ }

    // ... constructors, other methods,
    //    and other private data not shown
}
```

(a) Write the `School` method `getElectiveByName`. Method `getElectiveByName` should return the `Elective` in `electiveList` that has the given name.

Complete method `getElectiveByName` below.

```
// precondition: name is the name of an
//      Elective in electiveList
// postcondition: returns the Elective in
//      electiveList with the given name
private Elective getElectiveByName (String name)
```

GO ON TO THE NEXT PAGE.

(b) Write the `School` method `assignElectivesToStudents`. Method `assignElectivesToStudents` should assign electives to students as described at the beginning of this question.

In writing method `assignElectivesToStudents` you may use the `private` helper method `getElectiveByName` specified in part (a). Assume that `getElectiveByName` works as specified, regardless of what you wrote in part (a). Solutions that reimplement functionality provided by this method, rather than invoking itt, will not receive full credit.

Complete method `assignElectivesToStudents` below

```
// postcondition: All Students in
//       studentList have been either
//       assigned their first available
//       elective choice or not assigned;
//       All electives in electiveList have
//       been updated appropriately as
//       Students are assigned to them
public void assignElectivesToStudents ( )
```

GO ON TO THE NEXT PAGE.

(c) Write the `School` method `studentsWithoutElectives`. Method `studentsWithoutElectives` should return `ArrayList` of all Students in `studentList` who do not have an `Elective` assigned to them.

Complete method `studentsWithoutElectives` below

```
// postcondition: returns a list of those
//       Students who have not been assigned
//       an Elective
public ArrayList studentsWithoutElectives ( )
```

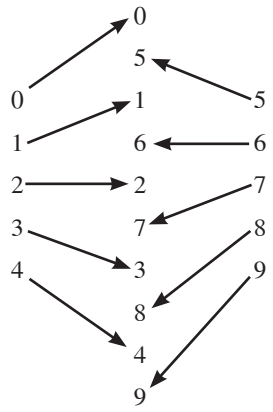
GO ON TO THE NEXT PAGE.

2. Consider a deck of n cards where n is even and each card is uniquely labeled from 1 to n.

   A *shuffle* is performed when the deck is divided into two stacks and the stacks are interlaced so that a new stack is formed by alternately taking cards from each stack.

   For instance, a deck of ten cards is in order when the card labeled 0 is on the top of the deck and the card labeled 9 is on the bottom of the deck.

   Dividing the deck in half produces two stacks of cards – one stack with cards 0 through 4, the other with cards 5 through 9. Interlacing the stacks produces a deck in the following order:



   The number of times needed to shuffle the deck until it returns to its original order is called the *reorder count*. Note that the reorder count for a deck of ten cards is six:

	Shuffle number					
Original	**1**	**2**	**3**	**4**	**5**	**6**
0	0	0	0	0	0	0
1	5	7	8	4	2	1
2	1	5	7	8	4	2
3	6	3	6	3	6	3
4	2	1	5	7	8	4
5	7	8	4	2	1	5
6	3	6	3	6	3	6
7	8	4	2	1	5	7
8	4	2	1	5	7	8
9	9	9	9	9	9	9

GO ON TO THE NEXT PAGE.

A deck is modeled by the following incomplete declaration of the **Deck** class:

```
public class Deck
{
    private int [ ] cards;

    public Deck (int numCards)
    { /* code not shown */ }

    public boolean inOrder ( )
    { /* to be implemented in part (a) */ }

    public void shuffle ( )
    { /* to be implemented in part (b) */ }

    public int reorderingCount ( )
    { /* to be implemented in part (c) */ }
}
```

(a) Write the Deck method inOrder. Method inOrder should return true if the cards in the deck are in numerical order from 0 to cards.length – 1 and should return false otherwise. Cards are in numerical order if cards [k] = = k for all 0 < = k < cards. length.

Complete method inOrder below.

```
// precondition: For all k such that
//       0 < = k < cards.length,
//       0 < = cards [k] < cards.length and
//       each cards [k] is unique
// postcondition: returns true if
//       cards [k] = = k for all
//       0 < = k < cards.length; returns
//       false otherwise
public boolean inOrder ( )
```

GO ON TO THE NEXT PAGE.

(b) Write the Deck method shuffle. Method shuffle should divide the deck into two equal stacks and interlace them evenly as described at the beginning of this question.

Complete method shuffle below.

```
// postcondition: the deck is shuffled by
//       dividing the deck into two equal stacks
//       that are evenly interlaced
public void shuffle ( )
```

GO ON TO THE NEXT PAGE.

(c) Write the Deck method reorderCount. Method reorderCount should return the number of shuffles necessary to return the deck to its original order.

In writing method reorderCount you may use the methods inOrder and shuffle as specified in parts (a) and (b). Assume that inOrder and shuffle work as specified, regardless of what you wrote in parts (a) and (b). Solutions that reimplement functionality provided by these methods, rather than invoking them, will not receive full credit.

Complete method reorderCount below.

```
// postcondition: returns the number of
//       shuffles necessary to return the cards
//       in the deck to their original numerical
//       order such that inOrder ( ) = = true; the
//       cards in the deck are in their original
//       numerical order
public int reorderCount ( )
```

GO ON TO THE NEXT PAGE.

3. Consider the design of an electronic cookbook modeled with the following class declarations:

```
public class Cookbook
{
    private ArrayList recipeList;
    // each entry is an instance of a
    // Recipe representing one recipe
    // in the cookbook
    // precondition: numPeople > 0
    // postcondition: All recipes in
    //     recipeList have been converted to
    //     serve numPeople number of people

    public void standardize (int numPeople)
    { /* code not shown */ }

    // ... constructors, other methods,
    //     and other private data not shown
}

public class Ingredient
{
    private String name;
    // the name of this ingredient

    private double amount;
    // the amount of this ingredient needed
    // in the recipe
    // postcondition: returns the amount of
    //     this ingredient needed in the
    //     recipe

    public double getAmount ( )
    { /* code not shown */ }
    // precondition: amt > 0.0
    // postcondition: amount has been set
    //     to amt

    public void setAmount (double amt)
    { /* code not shown */ }
    // precondition: newNumber > 0
    // postcondition: numberServed has
    // been set to newNumber

    public void setNumberServed(int newNumber)
    { /* code not shown */ }
    // ... constructors and other methods
    //     not shown
}
```

GO ON TO THE NEXT PAGE.

(a) A recipe in the cookbook is modeled by the class Recipe with the following data and operations:

Data

- return the list of ingredients
- the name of the recipe
- the list of ingredients used in the recipe
- the description of the preparation process for the recipe
- the number of people served by the recipe

Operations

- create a recipe with a given name and number of people served
- add an ingredient to the recipe
- set the description of the preparation process for the recipe
- return the name of the recipe
- return the number of people served by the recipe
- scale the recipe to serve a given new number of people by changing the amount of each ingredient appropriately

Write the definition of the class `Recipe`, showing the appropriate data definitions and constructor and method signatures. You should *not* write the implementations of the constructor or methods for the `Recipe` class.

(b) Using the signature you wrote in part (a), write the implementation for the method that scales the recipe to serve a given new number of people.

In writing this method, you may call any of the methods in the Recipe class (as you defined it in part (a)) or in the `Ingredient` class. Assume that these methods work as specified.

(c) Write the `Cookbook` method standardize as described at the beginning of the question.

In writing this method, you may call any of the methods in the Recipe class (as you defined it in part (a)). Assume that these methods work as specified.

Complete method standardize below.

```
// precondition: numPeople > 0
// postcondition: All recipes in
//       recipeList have been scaled to
//       serve numPeople number of people
public void standardize (int numPeople)
```

GO ON TO THE NEXT PAGE.

4. Consider a school that contains x number of students that all start their first period class in one of n classrooms. This scenario can be represented using three classes. The `School` class contains an array of all the `Classrooms` in the school. The `Classroom` class has fields for the teacher in charge of the room `teacherName` and an array of all the `Students` in the classroom `Students`. The `Student` class has a field for the name of the student `studentName` and the ID number of the student `studentID`.

   The `School` class contains a method `findStudent` that takes a teacher's name and a student ID as arguments and returns the name of the student. The method utilizes a sequential search algorithm to find the correct classroom and a binary search algorithm to find the correct student. If the student is not found in the school, the method returns "`Student Not Found`."

   Write the complete `School`, `Classroom`, and `Student` classes, including any instance variables, constructors, and necessary methods. You may assume that the student ID numbers in each classroom are sorted in ascending order.

STOP

END OF EXAM

The Princeton Review®

Completely darken bubbles with a No. 2 pencil. If you make a mistake, be sure to erase mark completely. Erase all stray marks.

1.

YOUR NAME: (Print) _____ Last _____ First _____ M.I.

SIGNATURE: _____ DATE: / /

HOME ADDRESS: (Print) _____ Number and Street

_____ City _____ State _____ Zip Code

PHONE NO.: _____

IMPORTANT: Please fill in these boxes exactly as shown on the back cover of your test book.

2. TEST FORM

3. TEST CODE

0	A	J	0	0
1	B	K	1	1
2	C	L	2	2
3	D	M	3	3
4	E	N	4	4
5	F	O	5	5
6	G	P	6	6
7	H	Q	7	7
8	I	R	8	8
9			9	9

4. REGISTRATION NUMBER

0	0	0	0	0	0
1	1	1	1	1	1
2	2	2	2	2	2
3	3	3	3	3	3
4	4	4	4	4	4
5	5	5	5	5	5
6	6	6	6	6	6
7	7	7	7	7	7
8	8	8	8	8	8
9	9	9	9	9	9

5. YOUR NAME

First 4 letters of last name					FIRST INIT	MID INIT
A	A	A	A		A	A
B	B	B	B		B	B
C	C	C	C		C	C
D	D	D	D		D	D
E	E	E	E		E	E
F	F	F	F		F	F
G	G	G	G		G	G
H	H	H	H		H	H
I	I	I	I		I	I
J	J	J	J		J	J
K	K	K	K		K	K
L	L	L	L		L	L
M	M	M	M		M	M
N	N	N	N		N	N
O	O	O	O		O	O
P	P	P	P		P	P
Q	Q	Q	Q		Q	Q
R	R	R	R		R	R
S	S	S	S		S	S
T	T	T	T		T	T
U	U	U	U		U	U
V	V	V	V		V	V
W	W	W	W		W	W
X	X	X	X		X	X
Y	Y	Y	Y		Y	Y
Z	Z	Z	Z		Z	Z

6. DATE OF BIRTH

Month	Day		Year	
JAN				
FEB	0	0	0	0
MAR	1	1	1	1
APR	2	2	2	2
MAY	3	3	3	3
JUN		4	4	4
JUL		5	5	5
AUG		6	6	6
SEP		7	7	7
OCT		8	8	8
NOV		9	9	9
DEC				

7. GENDER

- MALE
- FEMALE

The Princeton Review®

1.	A	B	C	D	E	11.	A	B	C	D	E	21.	A	B	C	D	E	31.	A	B	C	D	E
2.	A	B	C	D	E	12.	A	B	C	D	E	22.	A	B	C	D	E	32.	A	B	C	D	E
3.	A	B	C	D	E	13.	A	B	C	D	E	23.	A	B	C	D	E	33.	A	B	C	D	E
4.	A	B	C	D	E	14.	A	B	C	D	E	24.	A	B	C	D	E	34.	A	B	C	D	E
5.	A	B	C	D	E	15.	A	B	C	D	E	25.	A	B	C	D	E	35.	A	B	C	D	E
6.	A	B	C	D	E	16.	A	B	C	D	E	26.	A	B	C	D	E	36.	A	B	C	D	E
7.	A	B	C	D	E	17.	A	B	C	D	E	27.	A	B	C	D	E	37.	A	B	C	D	E
8.	A	B	C	D	E	18.	A	B	C	D	E	28.	A	B	C	D	E	38.	A	B	C	D	E
9.	A	B	C	D	E	19.	A	B	C	D	E	29.	A	B	C	D	E	39.	A	B	C	D	E
10.	A	B	C	D	E	20.	A	B	C	D	E	30.	A	B	C	D	E	40.	A	B	C	D	E

Practice Test 2

AP® Computer Science A Exam

SECTION I: Multiple-Choice Questions

DO NOT OPEN THIS BOOKLET UNTIL YOU ARE TOLD TO DO SO.

At a Glance

Total Time
1 hour 30 minutes
Number of Questions
40
Percent of Total Score
50%
Writing Instrument
Pencil required

Instructions

Section I of this examination contains 40 multiple-choice questions. Fill in only the ovals for numbers 1 through 40 on your answer sheet.

Indicate all of your answers to the multiple-choice questions on the answer sheet. No credit will be given for anything written in this exam booklet, but you may use the booklet for notes or scratch work. After you have decided which of the suggested answers is best, completely fill in the corresponding oval on the answer sheet. Give only one answer to each question. If you change an answer, be sure that the previous mark is erased completely. Here is a sample question and answer.

Sample Question

Chicago is a
(A) state
(B) city
(C) country
(D) continent
(E) county

Sample Answer

Use your time effectively, working as quickly as you can without losing accuracy. Do not spend too much time on any one question. Go on to other questions and come back to the ones you have not answered if you have time. It is not expected that everyone will know the answers to all the multiple-choice questions.

About Guessing

Many candidates wonder whether or not to guess the answers to questions about which they are not certain. Multiple-choice scores are based on the number of questions answered correctly. Points are not deducted for incorrect answers, and no points are awarded for unanswered questions. Because points are not deducted for incorrect answers, you are encouraged to answer all multiple-choice questions. On any questions you do not know the answer to, you should eliminate as many choices as you can, and then select the best answer among the remaining choices.

GO ON TO THE NEXT PAGE.

Java Quick Reference

Class Constructors and Methods	Explanation
String Class	
`String(String str)`	Constructs a new `String` object that represents the same sequence of characters as `str`
`int length()`	Returns the number of characters in a `String` object
`String substring(int from, int to)`	Returns the substring beginning at index `from` and ending at index `to - 1`
`String substring(int from)`	Returns `substring(from, length())`
`int indexOf(String str)`	Returns the index of the first occurrence of `str`; returns `-1` if not found
`boolean equals(String other)`	Returns `true` if `this` is equal to `other`; returns `false` otherwise
`int compareTo(String other)`	Returns a value <0 if `this` is less than `other`; returns zero if `this` is equal to `other`; returns a value >0 if `this` is greater than `other`
Integer Class	
`Integer(int value)`	Constructs a new `Integer` object that represents the specified `int` value
`Integer.MIN_VALUE`	The minimum value represented by an `int` or `Integer`
`Integer.MAX_VALUE`	The maximum value represented by an `int` or `Integer`
`int intValue()`	Returns the value of this `Integer` as an `int`
Double Class	
`Double(double value)`	Constructs a new `Double` object that represents the specified `double` value
`double doubleValue()`	Returns the value of this `Double` as a `double`
Math Class	
`static int abs(int x)`	Returns the absolute value of an `int` value
`static double abs(double x)`	Returns the absolute value of a `double` value
`static double pow(double base, double exponent)`	Returns the value of the first parameter raised to the power of the second parameter
`static double sqrt(double x)`	Returns the positive square root of a `double` value
`static double random()`	Returns a `double` value greater than or equal to `0.0` and less than `1.0`
ArrayList Class	
`int size()`	Returns the number of elements in the list
`boolean add(E obj)`	Appends `obj` to end of list; returns `true`
`void add(int index, E obj)`	Inserts `obj` at position `index` (`0 <= index <= size`), moving elements at position `index` and higher to the right (adds 1 to their indices) and adds 1 to size
`E get(int index)`	Returns the element at position `index` in the list
`E set(int index, E obj)`	Replaces the element at position `index` with `obj`; returns the element formerly at position `index`
`E remove(int index)`	Removes element from position `index`, moving elements at position `index + 1` and higher to the left (subtracts 1 from their indices) and subtracts 1 from size; returns the element formerly at position `index`
Object Class	
`boolean equals(Object other)`	
`String toString()`	

GO ON TO THE NEXT PAGE.

COMPUTER SCIENCE A

SECTION I

Time—1 hour and 30 minutes

Number of Questions—40

Percent of total exam grade—50%

Directions: Determine the answer to each of the following questions or incomplete statements, using the available space for any necessary scratchwork. Then decide which is the best of the choices given and fill in the corresponding oval on the answer sheet. No credit will be given for anything written in the examination booklet. Do not spend too much time on any one problem.

Notes:

- Assume that the classes listed in the Java Quick Reference have been imported where appropriate.
- Assume that declarations of variables and methods appear within the context of an enclosing class.
- Assume that method calls that are not prefixed with an object or class name and are not shown within a complete class definition appear within the context of an enclosing class.
- Unless otherwise noted in the question, assume that parameters in the method calls are not `null` and that methods are called only when their preconditions are satisfied.

1. Consider the following method.

```
public static int mystery(int a, int b)
{
   if (a <= 0)
        return b;
   else
        return mystery(a - 2, b);
}
```

What value is returned by the call `mystery(12, 5)`?

(A) `5`
(B) `6`
(C) `12`
(D) `60`
(E) `1565`

GO ON TO THE NEXT PAGE.

2. Consider the following instance variable and method.

```
private int[] numList;

//Precondition: numList contains a list of int values in no particular order

public int mystery(int n)
{
    for (int k = 0; k <= numList.length - 1; k++)
    {
       if (n <= numList[k])
          return k;
    }
    return numList.length;
}
```

Which of the following statements is most accurate about `numList` following the execution of the following statement?

```
int j = mystery(number);
```

(A) The greatest value in `numList` is at index `j`.
(B) The greatest value in `numList` that is less than `number` is at index `j`.
(C) All values in `numList` from index `0` to `j - 1` are greater than or equal to `number`.
(D) All values in `numList` from index `0` to `j - 1` are less than `number`.
(E) All values in `numList` from index `j` to `numList.length - 1` are greater than `number`.

GO ON TO THE NEXT PAGE.

Questions 3–4 refer to the following incomplete class declaration for a new data type called a Quack.

```
Public class Quack
{
   ArrayList<Object> myData;

   // Constructor initializes myData
   public Quack()
   {  /* implementation not shown */  }

   // Quack.
   public void enquack(Object x)
   {  /* implementation not shown */  }

   // if front is true, returns the object at the front end of the Quack; otherwise returns the object at the back end of
   the // Quack.  Assumes the Quack is not empty.
   public Object dequack(boolean front)
   {  /* implementation not shown */  }

   // Returns true if the Quack has no objects; otherwise returns false.
   public boolean isEmpty()
   {  /* implementation not shown */  }

   <designation> ArrayList myData;

   // ... other methods and data not shown
```

3. Which of the following is the best choice for <designation> and the best reason for that choice?

   (A) <designation> should be private so that programs using a Quack will not be able to modify myData by using methods enquack and dequack, thereby preserving the principle of data stability.

   (B) <designation> should be private so that programs using a Quack can modify myData only by using methods such as enquack and dequack, thereby preserving the principle of information hiding.

   (C) <designation> should be private as an indication to programs using a Quack that myData can be modified directly but that it is *better* to modify myData only by using methods such as enquack and dequack, thereby preserving the principle of maximum information dissemination.

   (D) <designation> should be public because programs using a Quack need to know how the Quack class has been implemented in order to use it.

   (E) <designation> should be public. Otherwise, only objects constructed from derived subclasses of a Quack will be able to modify the contents of a Quack.

4. Which of the following is an effective return statement for isEmpty as described in the incomplete declaration above?

   (A) `return (myData.length == 0)`
   (B) `return (size() == 0)`
   (C) `return (myData.size() == 0);`
   (D) `return (myData.length() == 0)`
   (E) `return (myData.size == 0)`

GO ON TO THE NEXT PAGE.

5. Consider the following method definition.

```
public static int mystery(int n)
{
   if (n <= 1)
      return 2;
   else
      return 1 + mystery(n - 3);
}
```

Which of the following lines of code can replace the line in `mystery` containing the recursive call so that the functionality of `mystery` does not change?

(A) `return 1 + ((n + 2) / 3);`
(B) `return 1 + ((n + 3) / 2);`
(C) `return 2 + ((n + 1) / 3);`
(D) `return 2 + ((n + 2) / 3);`
(E) `return 3 + ((n + 1) / 2);`

GO ON TO THE NEXT PAGE.

Questions 6–7 refer to the following incomplete class declaration.

```
public class DistanceTracker
{
    private int kilometers;
    private int meters;

    /* Constructs a DistanceTracker object
     * @param k the number of kilometers
     * Precondition: k ≥ 0
     * @param m the number of meters
     * Precondition: 0 ≤ m < 1000
     */
     public DistanceTracker (int k, int m)
     {
       kilometer = k;
       meters = m;
     }
     /* @return the number of kilometers
     */
     public int getKilometers()
     { /* implementation not shown */}
     /* @return the number of meters
     */
     public int getMeters()
     { /* implementation not shown */}
       /* Adds k kilometers and m meters
        * @param k the number of kilometers
        * Precondition: k ≥ 0
        *@param m the number of meters
     * Precondition: m ≥ 0
     */
     public void addDistance(int k, int m)
     {
       kilometers += k;
       meters += m;
       /* rest of method not shown */
     }
    //Rest of class not shown
    }
```

6. Which of the following code segments can be used to replace `/* rest of method not shown */` so `addDistance` will correctly increase the distance?

(A)
```
kilometers += meters / 1000
meters = meters % 1000
```
(B)
```
kilometers += meters % 1000
meters = meters / 1000
```
(C)
```
meters += kilometers % 1000
```
(D)
```
kilometers += meters % 1000
```
(E)
```
meters = meters % 1000
```

GO ON TO THE NEXT PAGE.

7. Consider the following incomplete class declaration.

```
public class DistanceTrackerSet
{
    Distance Tracker[] set;
    /*Declaration method not shown*/

    public  DistanceTracker total()
    {
       DistanceTracker temp = new DistanceTracker(0, 0);
    for (int k = 0; k < set.length; k++)
       {
          /* missing code */
       }
       return temp;
    }
/*Other methods not shown*/
}
```

Assuming `set` is properly initialized with `DistanceTracker` objects and all needed classes are properly imported, which of the following can be used to replace `/* missing code */` so that the method returns a `DistanceTracker` object with the total of all distances stored in `set`?

(A) `temp.addDistance(temp[k].kilometers, temp[k].meters);`
(B) `set[k].addDistance(temp[k].getKilometers(), temp[k].getMeters());`
(C) `set[k].addDistance();`
(D) `temp += temp.addDistance();`
(E) `temp.addDistance(temp[k].getKilometers(), temp[k].getMeters());`

8. Consider the following method.

```
public List<Integer> nums()
{
   List<Integer> values = new ArrayList<Integer>();
   for (int i = 0; i < 50; i = i + 5)
      if (i % 4 == 1)
         values.add(i);
   return value;
}
```

What will the return of `nums()` contain?

(A) `[5, 45]`
(B) `[5, 25, 45]`
(C) `[0, 20, 40]`
(D) `[5, 9, 13, 17, 21, 25, 29, 33, 37, 41, 45]`
(E) `[0, 5, 10, 15, 20, 25, 30, 35, 40, 45]`

GO ON TO THE NEXT PAGE.

9. Consider the following incomplete method `mystery`.

```
public static boolean mystery(boolean a, boolean b, boolean c)
{
   return <expression>;
}
```

What should `<expression>` be replaced with so that `mystery` returns `true` when exactly two of its three parameters are `true`; otherwise `mystery` returns `false`?

(A)
```
(a && b && !c) ||
(a && !b && c) ||
(!a && b && c)
```
(B)
```
(a && b && !c) &&
(a && !b && c) &&
(!a && b && c)
```
(C)
```
(a || b || !c) &&
(a || !b || c) &&
(!a || b || c)
```
(D)
```
(a && b) ||
(a && c) ||
(b && c)
```
(E)
```
(a || b) &&
(a || c) &&
(b || c)
```

10. Consider the following code segment.

```
int x;
x = 5 - 4 + 9 * 12 / 3 - 10;
```

What is the value of `x` after the code segment is executed?

(A) `13`
(B) `27`
(C) `30`
(D) `-57`
(E) `-10`

11. What is the best way to declare a variable `myStrings` that will store 50 `String` values if each `String` will be no longer than 25 characters?

(A) `ArrayList <String> myStrings[String[50]];`
(B) `ArrayList <String> myStrings = new String[50];`
(C) `ArrayList <String> myStrings = new String[25];`
(D) `String [] myStrings = new String [50, 25];`
(E) `String [] myStrings = new String [50];`

GO ON TO THE NEXT PAGE.

12. Consider the following code segment.

```
List <Integer> scores = new ArrayList<Integer>();
scores.add(93);
scores.add(97);
scores.add(84);
scores.add(91);
scores.remove(2);
scores.add(1, 83);
scores.set(3, 99);
System.out.println(scores);
```

What is the output of the code segment?

(A) `[83, 93, 99, 91]`
(B) `[93, 83, 91, 99]`
(C) `[83, 94, 91, 99]`
(D) `[93, 83, 97, 99]`
(E) The code throws an `ArrayIndexOutofBoundsException`.

13. Consider the following precondition, postcondition, and signature for the `getDigit` method.

```
// precondition: n >= 0
//               whichDigit >= 0
// postcondition: Returns the digit of n in
//                the whichDigit position
//                when the digits of n are
//                numbered from right to
//                left starting with zero.
//                Returns 0 if whichDigit >=
//                number of digits of n.
int getDigit (int n, int whichDigit)
```

Consider also the following three possible implementations of the `getDigit` method.

I.
```
if (whichDigit == 0)
  return n % 10;
else
  return getDigit(n / 10, whichDigit - 1);
```

II.
```
return (n / (int) Math.pow(10, whichDigit))  % 10;
```

III.
```
for (int k = 0; k < whichDigit; k++)
  n /= 10;
return n % 10;
```

Which implementation(s) would satisfy the postcondition of the `getDigit` method?

(A) I and II only
(B) I and III only
(C) II and III only
(D) I, II, and III
(E) None of the above

GO ON TO THE NEXT PAGE.

14. Consider an array of integers.

```
4    10    1    2    6    7    3    5
```

Assume that `SelectionSort` is used to order the array from smallest to largest values.

Which of the following represents the state of the array immediately after the first iteration of the outer `for` loop in the `SelectionSort` process?

(A) `1 4 10 2 3 6 7 5`
(B) `1 2 4 6 10 7 3 5`
(C) `1 10 4 2 6 7 3 5`
(D) `4 3 1 5 6 7 10 2`
(E) `5 3 7 6 2 1 10 4`

15. Assume that a program declares and initializes `v` as follows.

```
String [] v;
v = initialize ();              // Returns an array of
                                // length 10 containing
                                // ten valid strings
```

Which of the following code segments correctly traverses the array *backward* and prints out the elements (one per line)?

I.
```
for (int k = 9; k >= 0; k--)
    System.out.println(v[k]);
```

II.
```
int k = 0;
while (k < 10)
{
  System.out.println(v[9 - k]);
  k++;
}
```

III.
```
int k = 10;
while (k >= 0)
{
  System.out.println(v[k]);
  k--;
 }
```

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

GO ON TO THE NEXT PAGE.

16. Consider the following method.

```
/**Precondition: @param set an ArrayList that contains distinct integers
*@param n an int value
*/
public int mystery(List<Integer> set, int n)
{
    for (int k = 0; k < set.length(); k++)
    {
       if (set.get(k) > n)
       {
          return (set.remove(k) + mystery(set, n));
       }
    }
    return 0;
}
```

What is returned by the method call `mystery(set, n)`?

(A) `0`
(B) The number of the elements of `set` that are greater than `n`
(C) The sum of the elements of `set` that are greater than `n`
(D) The sum of the elements of `set` that are less than `n`
(E) The sum of the elements of `set` that are less than or equal to `n`

GO ON TO THE NEXT PAGE.

17. Consider the following two-dimensional array.

```
[ [0, 0, 0 ,0]
  [0, 1, 0, 0]
  [0, 1, 2, 0]
  [0, 1, 2, 3] ]
```

Which of the following methods returns this two-dimensional array?

(A)
```
public int[][] nums()
{
  int[][] temp = new int[4][4];
  for (int j = 0; j < 4; j++)
  {
      for (int k = 0; k < 4; k++)
      {
            temp[j][k] = j;
      }
  }
  return temp;
}
```

(B)
```
public int[][] nums()
{
  int[][] temp = new int[4][4];
  for (int j = 0; j < 4; j++)
  {
      for (int k = 0; k < 4; k++)
      {
            temp[j][k] = k;
      }
  }
  return temp;
}
```

(C)
```
public int[][] nums()
{
  int[][] temp = new int[4][4];
  for (int j = 0; j < 4; j++)
  {
      for (int k = j; k < 4; k++)
      {
            temp[j][k] = k;
      }
  }
  return temp;
}
```

(D)
```
public int[][] nums()
{
  int[][] temp = new int[4][4];
  for (int j = 0; j < 4; j++)
  {
      for (int k = 0; k <= j; k++)
      {
            temp[j][k] = j;
      }
  }
  return temp;
}
```

(E)
```
public int[][] nums()
{
  int[][] temp = new int[4][4];
  for (int j = 0; j < 4; j++)
  {
      for (int k = 0; k <= j; k++)
      {
            temp[j][k] = k;
      }
  }
  return temp;
}
```

GO ON TO THE NEXT PAGE.

18. A children's club classifies members based on age according to the table below:

Years	Classification
Under 3	Infant
3 to 7 inclusive	Pee-Wee
8 to 13 inclusive	Cub
Over 14	Leader

Which of the following methods will correctly take the integer parameter `age` and return the string `Classification`?

(A)
```
public String Classification(int age)
{
   String temp;
   if (age < 3)
      temp = "Infant";
   if (age <= 7)
      temp = "Pee-Wee";
   if (age <= 13)
      temp = "Cub";
   if (age >= 14)
      temp = "Leader";
   return temp;
}
```

(B)
```
public String Classification(int age)
{
   String temp;
   if (age < 3)
      temp = "Infant";
   if (3 <= age <= 7)
      temp = "Pee-Wee";
   if (8 <= age <= 13)
      temp = "Cub";
   if (age >= 14)
      temp = "Leader";
   return temp;
}
```

(C)
```
public String Classification(int age)
{
   String temp;
   if (age < 3)
      temp = "Infant";
   else if (age <= 7)
      temp = "Pee-Wee";
   else if (age <= 13)
      temp = "Cub";
   else if (age >= 14)
      temp = "Leader";
   return temp;
}
```

(D)
```
public String Classification(int age)
{
   String temp;
   if (age < 3)
      temp = "Infant";
   else if (age < 7)
      temp = "Pee-Wee";
   else if (age < 13)
      temp = "Cub";
   else if (age > 14)
      temp = "Leader";
   return temp;
}
```

(E)
```
public String Classification(int age)
{
   String temp;
   if (age < 3)
      temp = "Infant";
   if (age < 7)
      temp = "Pee-Wee";
   if (age < 13)
      temp = "Cub";
   if (age > 14)
      temp = "Leader";
   return temp;
}
```

GO ON TO THE NEXT PAGE.

Questions 19–20 refer to the method `getGap` with line numbers added for reference. Method `getGap` is intended to find the maximum difference between the indices of any two occurrences of `num` in the array `arr`. However, the method `getGap` does not work as intended.

For example, if the array `arr` contains `[8, 7, 5, 5, 4, 7, 2, 7, 1, 2, 7]`, the call `getGap(arr, 7)` should return `9`, the difference between the indices of the first and last occurrences of `7`.

/**Precondition: `arr` contains at least two occurrences of `num` */

```
1     public int getGap(int[] arr, int num)
2     {
3          int index1 = -1;
4          int index2 = -1;
5          for (int k = 0; k < arr.length; k++)
6          {
7               if (arr[k] == num)
8               {
9                    if (index1 == -1)
10                   {
11                        index1 = k;
12                        index2 = k;
13                   }
14                   else
15                   {
16                        index1 = index2;
17                        index2 = k;
18                   }
19              }
20         }
21         return (index2 - index1);
22    }
```

GO ON TO THE NEXT PAGE.

19. The method `getGap` does not work as intended. Which of the following best describes the return of the method `getGap` ?

    (A) The difference between the indices of the last two occurrences of `num` in `arr`
    (B) The minimum difference between the indices of any two occurrences of `num` in `arr`
    (C) The difference between the indices of the first two occurrences of `num` in `arr`
    (D) The length of the array `arr`
    (E) The number of occurrences of `num` in `arr`

20. Which of the following changes should be made to `getGap` so that the method will work as intended?

    (A) Delete the statement at line 4.
    (B) Delete the statement at line 11.
    (C) Delete the statement at line 12.
    (D) Delete the statement at line 16.
    (E) Delete the statement at line 17.

GO ON TO THE NEXT PAGE.

Questions 21–23 refer to the following incomplete class declaration that is intended to be used to represent calendar dates.

```
Public class Date
{
   private int month;
       // represents month 0–11
   private int day;
       // represents day of the month
       // 0–31
   private int year;
       // represents the year

   // constructor sets the private data
   public Date (int m, int d, int y)
   { /* implementation not shown */ }

   // postconditions: returns the month
   public int getMonth()
   { /* implementation not shown */ }

   // postcondition: return the day
   public int getDay()
   { /* implementation not shown */ }

   // postcondition: returns the year
   public int getYear()
   { /* implementation not shown */ }

   // postcondition: returns the number of
   //                   days which, when
   //                   added to this Date,
   //                   gives newDate
   public int daysUntil (Date newDate)
   { /* implementation not shown */ }

   // postcondition: returns true if
   //                   the month, day, and
   //                   year of this Date are
   //                   are equal to those of
   //                   other; otherwise
   //                   returns false
   public boolean equals (Date other)
   { /* implementation not shown */ }

   // ... other methods not shown
```

GO ON TO THE NEXT PAGE.

21. Consider the method `equals` of the `Object` class.

    Which of the following method signatures is appropriate for the `equals` method?

    (A) `public boolean equals (Object other)`
    (B) `public int equals (Object other)`
    (C) `public boolean equals (Date other)`
    (D) `public int equals (Date other)`
    (E) `public boolean equals (Date d1, Date d2)`

22. Which of the following code segments could be used to implement the `equals` method of the `Date` class so that the `equals` method works as intended?

    I.
    ```
    if (month == other.month)
      if (day == other.day)
            if (year == other.year)
                  return true;
    Return false;
    ```
    II.
    ```
    if (month == other.getMonth() &&
            day == other.getDay() &&
            year == other.getYear())
      return true;
    else
      return false;
    ```
    III.
    ```
    return !((getMonth() != other.getMonth())  ||
      (getDay() != other.getDay())  ||
      (getYear() != other.getYear()));
    ```

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

23. During the testing of the `Date` class, it is determined that the class does not correctly handle leap years—although it handles non-leap years correctly.

    In which method of the `Date` class is the problem most likely to be found?

    (A) The `Date` constructor
    (B) The `getMonth` method
    (C) The `getDay` method
    (D) The `daysUntil` method
    (E) The `equals` method

GO ON TO THE NEXT PAGE.

24. Consider the following methods:

```
public static void mystery()
{
   int [] A;
   A = initialize ();
       // returns a valid initialized
       // array of integers
   for (int k = 0; k < A.length / 2; k++)
      swap (A[k], A[A.length - k - 1]);
}

public static void swap (int x, int y)
{
   int temp;
   temp = x;
   x = y;
   y = temp;
}
```

Which of the following best characterizes the effect of the `for` loop in the method `mystery`?

(A) It sorts the elements of `A`.
(B) It reverses the elements of `A`.
(C) It reverses the order of the first half of `A` and leaves the second half unchanged.
(D) It reverses the order of the second half of `A` and leaves the first half unchanged.
(E) It leaves all of the elements of `A` in their original order.

GO ON TO THE NEXT PAGE.

25. Consider the following code segment:

```
int [][] A = new int [4][3];
for (int j = 0; j < A[0].length; j++)
   for (int k = 0; k < A.length; k++)
      if (j == 0)
           A[k][j] = 0;
      else if (k % j == 0)
           A[k][j] = 1;
      else
           A[k][j] = 2;
```

What are the contents of `A` after the code segment has been executed?

(A)
```
0 0 0 0
1 1 1 1
1 2 1 2
```
(B)
```
0 1 1 1
0 2 2 2
0 1 2 1
```
(C)
```
0 0 0
1 1 2
1 1 1
1 1 2
```
(D)
```
0 1 1
0 2 1
0 2 2
0 2 1
```
(E)
```
0 1 1
0 1 2
0 1 1
0 1 2
```

26. Consider the following method:

```
/** @param num an int value such that num >= 0
 */
public void mystery(int num)
{
   System.out.print(num % 100);
   if ((num / 100) != 0)
   {
      mystery(num / 100);
   }
   System.out.print(num % 100);
}
```

Which of the following is printed as a result of the call `mystery(456789)`?

(A) `456789`
(B) `896745`
(C) `987654`
(D) `456789896745`
(E) `896745456789`

GO ON TO THE NEXT PAGE.

27. Consider the following method:

```
public static int mystery(int x, int y)
{
   if (x > 0)
      return x;
   else if (y > 0)
      return y;
   else
      return x / y;
}
```

In accordance with good design and testing practices, which of the following is the best set of test cases (`x, y`) for the method `mystery`?

(A) `(3, 4), (-3, 4), (-3, -4)`
(B) `(3, 4), (-3, 4), (-3, -4), (-3, 0)`
(C) `(3, 4), (3, -4), (-3, -4), (-3, 0)`
(D) `(3, 4), (3, -4), (3, 0), (-3, 4), (-3, -4)`
(E) `(3, 4), (2, 5), (3, -4), (-3, 0), (4, 0), (0, 0)`

28. Consider the following method.

```
/** Precondition: numList is not empty
 */
private int mystery(int[] numList)
{
   int n = numList[numList.length - 1];
   for (int k : numList)
   {
    if (n > k)
       {
        n = k;
       }
   }
   return n;
}
```

Which of the following best describes the return of `mystery`?

(A) The largest value in the array `numList`
(B) The least value in the array `numList`
(C) The index of the largest value in the array `numList`
(D) The index of the least value in the array `numList`
(E) The number of indices whose values are less than `numList[n]`

GO ON TO THE NEXT PAGE.

29. Consider the following method.

```
public int[] editArray(int[] arr, int oldNum, int newNum)
{
    /* missing code */
    return arr;
}
```

The method above is intended to replace any instances of `oldNum` in `arr` with instances of `newNum`. Which of the following can be used to replace `/* missing code */` to replace any values of `oldNum` in the array with values of `newNum`?

(A)
```
for (int k = 0; k < arr.length; k++)
{
   if (arr[k] = oldNum)
   {
      arr[k] == newNum;
   }
}
```

(B)
```
for (int k = 0; k < arr.length; k++)
{
   if (arr[k] == oldNum)
   {
      arr[k] = newNum;
   }
}
```

(C)
```
while (arr[k] == oldNum)
{
   arr[k] = newNum
}
```

(D)
```
for (int k = 0; k < arr.length; k++)
{
   arr[k] == newNum;
}
```

(E)
```
while (int k = 0; k < arr.length; k++)
{
   if (arr[k] = oldNum)
   {
      arr[k] == newNum;
   }
}
```

GO ON TO THE NEXT PAGE.

30. Consider the following two classes.

```
public class SalesPerson
{
    public void sale()
    {
       System.out.print("greet ");
       pitch();
    }
    public void pitch()
    {
       System.out.print("pitch ");
    }

}

public class CommissionedSalesPerson extends SalesPerson
{
    public void sale()
    {
       super.sale();
       System.out.print("record ");
    }
    public void pitch()
    {
       super.pitch();
       system.out.print("close ");
    }

}
```

The following code segment is found in a class other than `SalesPerson`.

```
SalesPerson vincent = new CommissionedSalesPerson();
vincent.sale();
```

Which of the following is printed as a result of this code segment?

(A) `greet pitch`
(B) `greet pitch close`
(C) `greet pitch record`
(D) `greet pitch record close`
(E) `greet pitch close record`

GO ON TO THE NEXT PAGE.

31. Consider the following declaration of a class that will be used to represent dimensions of rectangular crates.

```
public class Crate
{
    private int length;
    private int width;
    private int height;

    public Crate(int x, int y, int z)
    {
       length = x;
       width = y;
       height = z;
    }

    //other methods not shown
}
```

The following incomplete class declaration is intended to extend the `Crate` class so that the color of the crate can be specified.

```
public class ColoredCrate extends Crate
{
   private String color;
   //Constructors not shown
   //Other methods not shown
}
```

Which of the following possible constructors for `ColoredCrate` would be considered legal?

I.
```
public ColoredCrate(int a, int b, int c, String crateColor)
{
   length = a;
   width = b;
   height = c;
   color = crateColor;
}
```

II.
```
public ColoredCrate(int a, int b, int c, String crateColor)
{
   super (a, b, c);
   color = crateColor;
}
```

III.
```
public ColoredCrate()
{
   color = "";
}
```

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

GO ON TO THE NEXT PAGE.

32. Consider the following three proposed implementations of method `reverse`, intended to return an `ArrayList` of the objects in reversed order:

I.
```
public static ArrayList<Object> reverse (ArrayList<Object> q)
{
      ArrayList<Object> s = new ArrayList<Object>();
      while (q.size() != 0)
            s.add(0, q.remove(0));
      return s;
}
```

II.
```
public static ArrayList<Object> reverse (ArrayList<Object> q)
{
      ArrayList<Object> s = new ArrayList<Object>();
      for (int k = 0; k < q.size(); k++)
            s.add(0, q.remove(0));
      return s;
}
```

III.
```
public static ArrayList<Object> reverse (ArrayList<Object> q)
{
      Object obj;
      if (q.size() != 0)
      {
            obj = q.remove(0);
            q = reverse(q);
            q.add(obj);
      }
      return q;
}
```

Which of the above implementations of method `reverse` work as intended?

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

GO ON TO THE NEXT PAGE.

33. Consider the following code segment.

```
List<Integer> values = new ArrayList<Integer>();
values.add(5);
values.add(3);
values.add(2);
values.add(2);
values.add(6);
values.add(3);
values.add(9);
values.add(2);
values.add(1);
for (int j = 0; j < values.size(); j++)
{
    if (values.get(j).intValue() == 2)
    {
       values.remove(j);
    }
}
```

What will `values` contain as a result of executing this code segment?

(A) `[5, 3, 2, 2, 6, 3, 9, 2, 1]`
(B) `[5, 3, 2, 6, 3, 9, 1]`
(C) `[5, 3, 6, 3, 9, 1]`
(D) `[2, 2, 2, 5, 3, 6, 3, 9, 1]`
(E) The code throws an `ArrayIndexOutOfBoundsException`.

GO ON TO THE NEXT PAGE.

34. Consider the class `Data` partially defined below. The completed `max1D` method returns the maximum value of `b`, a one-dimensional array of integers. The completed `max2D` method is intended to return the maximum value of `c`, a two-dimensional array of integers.

```
public class Data
{
    /** Returns the maximum value of one-dimensional array b */
    public int max1D(int[] b)
    { /* implementation not shown */}
    /** Returns the maximum value of two-dimensional array c */
    public int max2D(int[] c)
    {
        int max;
        /* missing code */
        returns max
    }
    /* other methods of Data class not shown*/
}
```

Assume that `max1D` works as intended. Which of the following can replace `/* missing code */` so that `max2D` works as intended?

I.
```
for (int[] row: c)
{
    max = max1D(row);
}
```

II.
```
max = max1D(c[0]);
for (int k = 1; k <= c.length; k++)
{
    max = max1D(c[k]);
}
```

III.
```
max = max1D(c[0]);
for (int[] row: c)
{
    if (max < maxID(row))
    {
        max = max1D(row);
    }
}
```

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

GO ON TO THE NEXT PAGE.

35. Consider the following instance variable, `numList`, and incomplete method, `countZeros`. The method is intended to return an integer array `count` such that for all `k`, `count[k]` is equal to the number of elements equal to `0` from `numList[0]` through `numList[k]`. For example, if `numList` contains the values `{1, 4, 0, 5, 0, 0}`, the array `countZeros` contains the values `{0, 0, 1, 1, 2, 3}`.

```
public int[] countZeros(int[] numList)
{
   int[] count = new int[numList.length];
   for (int k : count)
   {
      count[k] = 0;
   }
   /* missing code */
   return count;
}
```

The following two versions of `/* missing code */` are suggested to make the method work as intended.

Version 1

```
for (int k = 0;  k <= numList.length; k++)
{
   for (int j = 0; j <= k; j++)
   {
        if  (numList[j] == 0)
        {
             count[k] = count[k] + 1;
        }
   }
}
```

Version 2

```
for (int k = 0; k < numList.length; k++)
{
   if  (numList[k] = 0)
   {
        count[k] = count[k - 1] + 1;
   }
   else
   {
        count[k] = count[k - 1];
   }
}
```

Which of the following statements is true?

(A) Both Version 1 and Version 2 will work as intended, but Version 1 is faster than Version 2.
(B) Both Version 1 and Version 2 will work as intended, but Version 2 is faster than Version 1.
(C) Version 1 will work as intended, but Version 2 causes an `ArrayIndexOutOfBoundsException`.
(D) Version 2 will work as intended, but Version 1 causes an `ArrayIndexOutOfBoundsException`.
(E) Version 1 and Version 2 each cause an `ArrayIndexOutOfBoundsException`.

GO ON TO THE NEXT PAGE.

36. A real estate agent wants to develop a program to record information about apartments for rent. For each apartment, she intends to record the number of bedrooms, number of bathrooms, whether pets are allowed, and the monthly rent charged. Which of the following object-oriented program designs would be preferred?

(A) Use a class `Apartment` with four subclasses: `Bedrooms`, `Bathrooms`, `PetsAllowed`, and `Rent`.
(B) Use four classes: `Bedrooms`, `Bathrooms`, `PetsAllowed`, and `Rent`, each with subclass `Apartment`.
(C) Use a class `Apartment` with four instance variables `int bedrooms`, `int bathrooms`, `boolean petsAllowed`, and `double rent`.
(D) Use five unrelated classes: `Apartment`, `Bedrooms`, `Bathrooms`, `PetsAllowed`, and `Rent`.
(E) Use a class `Apartment`, with a subclass `Bedrooms`, with a subclass `Bathrooms`, with a subclass `PetsAllowed`, with a subclass `Rent`.

37. Consider the following declarations:

```
public class Book
{
    boolean hasMorePagesThan(Book b);
    //other methods not shown
}
Public class Dictionary extends Book
{
    //other methods not shown
}
```

Of the following method headings of `hasMorePagesThan`, which can be added to `Dictionary` so that it will satisfy the `Book` superclass?

I. `int hasMorePagesThan(Book b)`
II. `boolean hasMorePagesThan(Book b)`
III. `boolean hasMorePagesThan(Dictionary d)`

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

GO ON TO THE NEXT PAGE.

38. Consider the following method.

```
/**Precondition: set does not contain any negative values
 */
public int mystery(int[] set, int max)
{
    int m = 0;
    int count = 0;
    for (int n = 0; n < set.length && set[n] < max; n++)
    {
        if (set[n] >= m)
        {
              m = set[n]; //Statement A
        }
        count++; //Statement B
    }
    return count;
}
```

Assume that `mystery` is called and is executed without error. Which of the following are possible combinations of the value of `max`, the number of times Statement A is executed, and the number of times Statement B is executed?

	Value of max	Executions of Statement A	Executions of Statement B
I.	8	2	3
II.	3	7	5
III.	7	0	4

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

GO ON TO THE NEXT PAGE.

39. The following method is intended to return an array that inserts an integer `m` at index `n`, pushing the values of all indices after `n` to the index one higher. For example, if the `arr` is

4	2	1	3

and the command `arr = insert(arr, 5, 2)` is called, the method is intended to return

4	2	5	1	3

```
1         public int[] insert(int[] arr, int m, int n)
2         {
3                int[] temp = new int[arr.length + 1];
4                for (int k  = 0; k < arr.length; k++)
5                {
6                       if (k < n)
7                       {
8                              temp[k] = arr[k];
9                       }
10                      else
11                      {
12                             temp[k + 1] = arr[k];
13                      }
14               }
15               temp[m] = n;
16               return temp;
17        }
```

The method `insert` does not work as intended. Which of the following changes will cause it to work as intended?

(A) Change Line 6 to `if (k > n)`
(B) Change Line 6 to `if (k <= n)`
(C) Change Line 12 to `temp[k] = arr[k + 1];`
(D) Change Line 15 to `temp[n] = m;`
(E) Change Line 16 to `return arr;`

GO ON TO THE NEXT PAGE.

40. If X, Y, and Z are integer values, the boolean expression

```
(X > Y)  &&  (Y  >  Z)
```

can be replaced by which of the following?

(A) `X  >  Z`

(B) `(X  <  Y)  ||  (Y  <  Z)`

(C) `(X <= Y)  ||  (Y  <= Z)`

(D) `!((X  <  Y)  ||  (Y  <  Z))`

(E) `!((X  <=  Y)  ||  (Y  <=  Z))`

END OF SECTION I

IF YOU FINISH BEFORE TIME IS CALLED,
YOU MAY CHECK YOUR WORK ON THIS SECTION.

DO NOT GO ON TO SECTION II UNTIL YOU ARE TOLD TO DO SO.

COMPUTER SCIENCE A

SECTION II

Time—1 hour and 30 minutes

Number of Questions—4

Percent of Total Grade—50%

Directions: SHOW ALL YOUR WORK. REMEMBER THAT PROGRAM SEGMENTS ARE TO BE WRITTEN IN JAVA™.

Notes:

- Assume that the classes listed in the Java Quick Reference have been imported where appropriate.
- Unless otherwise noted in the question, assume that parameters in method calls are not `null` and that methods are called only when their preconditions are satisfied.
- In writing solutions for each question, you may use any of the accessible methods that are listed in classes defined in that question. Writing significant amounts of code that can be replaced by a call to one of these methods will not receive full credit.

FREE-RESPONSE QUESTIONS

1. A binary, or base two, integer is a number consisting of digits that are either 0 or 1. Digits in a binary integer are numbered from right to left starting with 0.

   The decimal value of the binary integer is the sum of each digit multiplied by 2^d where d is the number of the digit.

   For example, the decimal value of the binary integer 1011010 is

   $$(0 * 2^0) + (1 * 2^1) + (0 * 2^2) + (1 * 2^3) + (1 * 2^4) + (0 * 2^5) + (1 + 2^6)$$

   $$= 0 + 2 + 0 + 8 + 16 + 0 + 64$$

   $$= 90$$

   A decimal integer can be converted into its corresponding binary integer according to the following algorithm:

   - Calculate the remainder when the decimal integer is divided by 2. This is the rightmost digit of the corresponding binary integer.
   - Divide the decimal integer by 2 using integer division. If the result is 0, stop. Otherwise, repeat the algorithm using the new value of the decimal integer.

   The digits produced will be in right-to-left order in the binary integer.

   For instance, the decimal integer 90 can be converted into its corresponding binary integer as follows:

	90 % 2 = 0	(the rightmost digit)
90 / 2 = 45	45 % 2 = 1	(the second digit from the right)
45 / 2 = 22	22 % 2 = 0	(the third digit from the right)
22 / 2 = 11	11 % 2 = 1	(the fourth digit from the right)
11 / 2 = 5	5 % 2 = 1	(the fifth digit from the right)
5 / 2 = 2	2 % 2 = 0	(the sixth digit from the right)
2 / 2 = 1	1 % 2 = 1	(the leftmost digit)
1 / 2 = 0		

GO ON TO THE NEXT PAGE.

(a) Write the method `Convert to Binary`. Method `Convert to Binary` should convert an integer decimal `Number` to a `String` representation of that number in binary. Return the binary representation.

```
//precondition: 0 < = decimal Number < = 2147483647

//postcondition: returns the string representation of the binary representation

public static String Convert to Binary (int decimal Number)
```

Begin your response at the top of a new page in the separate Free Response booklet and fill in the appropriate circle at the top of each page to indicate the question number. If there are multiple parts to this question, write the part letter with your response.

GO ON TO THE NEXT PAGE.

(b) Write the method `Convert to Decimal`. Method `Convert to Decimal` should convert the string representation of a binary number to an integer. The integer should be returned.

```
public static String Convert to Decimal (String binary Number).
```

Begin your response at the top of a new page in the separate Free Response booklet and fill in the appropriate circle at the top of each page to indicate the question number. If there are multiple parts to this question, write the part letter with your response.

GO ON TO THE NEXT PAGE.

2. A parabola is a graph defined by the equation $y = ax^2 + bx + c$, where a, b, and c are all integers and a is non-zero. The x-value of the axis of symmetry of a parabola is defined by the double $-b/2a$. A point is a pair of integer values, x and y. A point is defined to be on a parabola if it satisfies the equation of the parabola. Consider the examples in the table below:

Equation	Axis of symmetry ($-b/2a$)	Is point (4, 3) on the parabola?
$y = 2x^2 - 6x - 5$	$-(-6)/2(2) = 1.5$	Yes, $3 = 2(4)^2 - 6(4) - 5$
$y = 4x^2 + 2x - 3$	$-2/2(4) = -0.25$	No, $3 \neq 4(4)^2 + 2(4) - 3$

The following code segment is from a method outside the class `Parabola` and demonstrates how the `Parabola` class can be used to represent the two equations above:

```
Parabola par1 = new Parabola(2, -6, -5);
double axis1 = par1.getAxis(); //assigns 1.5 to axis1
boolean onGraph1 = par1.isOnGraph(4, 3); //assigns true to onGraph1

Parabola par2 = new Parabola(4, 2, -3);
double axis2 = par2.getAxis(); //assigns -0.25 to axis2
boolean onGraph2 = par2.isOnGraph(4, 3); //assigns false to onGraph2
```

Write the `Parabola` class. The constructor class of `Parabola` must take three integer parameters that represent a, b, and c, successively. You may assume as a precondition that a be a non-zero integer. You must include a `getAxis` method that returns the x-coordinate of the axis of symmetry as a double and an `isOnGraph` method that takes a point represented by two integer parameters, x and y, and returns `true` if the point is on the `Parabola` and returns `false` if it is not. Your class methods must be able to return the values indicated in the examples above. You can ignore any overflow issues.

Begin your response at the top of a new page in the separate Free Response booklet and fill in the appropriate circle at the top of each page to indicate the question number. If there are multiple parts to this question, write the part letter with your response.

GO ON TO THE NEXT PAGE.

3. A toy collector is creating an inventory of her marble collection. A marble set specifies the color and number of a particular group of marbles from her collection. The declaration of the `MarbleSet` class is shown below.

```
public class MarbleSet
{
    /** Constructs a new MarbleSet object */
    public MarbleSet(String color, int numMarbles)
    {  /* implementation not shown */  }

    /** @return the color of the set of marbles
     */
    public String getColor()
    {  /* implementation not shown */  }

    /** @return the number of marbles in the set
      */
     public int getNumber()
     {  /* implementation not shown */  }

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

The `MarbleCollection` class documents all sets of marbles in the collection. The declaration of the `MarbleCollection` class is shown below.

```
public class MarbleCollection
{
    /** This is a list of all marble sets */
    private List<MarbleSet> sets;

    /** Constructs a new MarbleSet object */
    public MarbleCollection()
    {  sets = new ArrayList<MarbleSet>();      }

    /** Adds theSet to the marble collection
     * @param theSet the marble set to add to the marble collection
     */
    public void addSet(MarbleSet theSet)
    {  sets.add(theSet);    }

    /** @return the total number of marbles
      */
     public int getTotalMarbles()
     {  /* to be implemented in part (a) */  }
    public int removeColor(String marbleColor)
     {  /* to be implemented in part (b) */  }
    /** Removes all the marble sets from the marble collection that have the same color as
     * marbleColor and returns the total number of marbles removed
     * @param marbleColor the color of the marble sets to be removed
     * @return the total number of marbles of marbleColor in the marble sets removed
     */
```

GO ON TO THE NEXT PAGE.

(a) The `getTotalMarbles` method computes and returns the sum of the number of marbles. If there are no marble sets, the method returns `0`.

```
Complete method  getTotalMarbles below
/** @return the sum of the number of marbles in all marble sets
 */
public int getTotalMarbles()
```

Begin your response at the top of a new page in the separate Free Response booklet and fill in the appropriate circle at the top of each page to indicate the question number. If there are multiple parts to this question, write the part letter with your response.

GO ON TO THE NEXT PAGE.

(b) The method `removeColor` updates the marble collection by removing all the marble sets for which the color of the marbles matches the parameter `marbleColor`. The marble collection may contain zero or more marbles with the same color as the `marbleColor`. The method returns the number of marbles removed.

For example, after the execution of the following code segment

```
MarbleCollection m = new MarbleCollection();
m.addSet(new MarbleSet("red", 2);
m.addSet(new MarbleSet("blue", 3);
m.addSet(new MarbleSet("green", 3);
m.addSet(new MarbleSet("blue", 4);
m.addSet(new MarbleSet("red", 1);
```

the contents of the marble collection can be expressed with the following table.

"red" 2	"blue" 3	"green" 3	"blue" 4	"red" 1

The method call `m.removecolor("red")` returns `3` because there were two red marble sets containing a total of `3` marbles. The new marble collection is shown below.

"blue" 3	"green" 3	"blue" 4

The method call `m.removecolor("purple")` returns `0` and makes no modifications to the marble collection.

Complete the method `removeColor` below.

```
/** Removes all the marble sets from the marble collection that have the same
*color as marbleColor and returns the total number of marbles removed
* @param marbleColor the color of the marble sets to be removed
* @return the total number of marbles of marbleColor in the marble sets removed
*/
      public int removeColor(String marbleColor)
```

Begin your response at the top of a new page in the separate Free Response booklet and fill in the appropriate circle at the top of each page to indicate the question number. If there are multiple parts to this question, write the part letter with your response.

GO ON TO THE NEXT PAGE.

4. Connect 4 is a popular checker game in which two players alternate taking turns, attempting to place four of their checkers in a straight line to win. One player will be represented by black checkers; the other player will be represented by white checkers. Their moves will be represented in a two-dimensional `String` array using "B" for black and "W" for white.

   The class `Connect4`, shown below, uses a two-dimensional array to represent the playing board.

```
public class Connect4
{
/** board[r][c] represents the color of the checker in row r and column c on the board. */

    private String[][] board;

    /** Creates a board with the given number of rows and columns
     *  @param rows - the rows on the board
     *  @param cols - the columns on the board
     *  Precondition: rows >= 4, cols >= 4
     *  Postcondition - elements in the array are initialized to null
     */
    Connect4(int rows, int cols)
    {  /* to be implemented in part (a) */}

    /**  Will initialize the lowest row in the board available for a specified column
     * @param color  - the color of the checker
     * @param column - the column to initialize the checker,
     *               - returns true if there is an available element to initialize,
     *                 otherwise returns false
     * Precondition:  - color will be "B" or "W"
     * Postcondition: - if a null element in the specified column exists, the appropriate
     *            element in the board will be updated with color, true will be returned
     *           - if a null element in the specified column does not exist,
     *           false will be returned
     *
     */
    public boolean move(String color, int column)
    {  /* to be implemented in part (b) */ }

    /**  Will check the each column to determine if four consecutive checkers of the
     *   specified color exist
     *  @param color  - the color of the checker
     *
     * Precondition:  - color will be "B" or "W"
     * Postcondition:  the index of four consecutive checkers of the specified color is
     *                  returned, otherwise, -1 is returned
     */
    public int checkVerticalWin(String color)

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

GO ON TO THE NEXT PAGE.

(a) Write the constructor for the `Connect4` class. The constructor initializes the `board` instance variable to a two-dimensional array with the given number of rows and columns.

`Connect4(rows, cols)` will construct new board. A call of `Connect4(7, 9)` would create the board as illustrated below.

	0	1	2	3	4	5	6	7	8
0	null	null	null	null	null	null	null	null	null
1	null	null	null	null	null	null	null	null	null
2	null	null	null	null	null	null	null	null	null
3	null	null	null	null	null	null	null	null	null
4	null	null	null	null	null	null	null	null	null
5	null	null	null	null	null	null	null	null	null
6	null	null	null	null	null	null	null	null	null

```
Connect4(rows, cols)
```

Begin your response at the top of a new page in the separate Free Response booklet and fill in the appropriate circle at the top of each page to indicate the question number. If there are multiple parts to this question, write the part letter with your response.

GO ON TO THE NEXT PAGE.

(b) Write the `move` method for the `Connect4` class which initializes an element in the board with the specified `color`, at the lowest possible row of the specified `column`. Return `true` if a color has been successfully assigned to the column, `false` if not.

The following 2 calls would have the result shown below.

```
move("B", 6)
move("W", 7)
```

	0	1	2	3	4	5	6	7	8
0	null	null	null	null	null	null	null	null	null
1	null	null	null	null	null	null	null	null	null
2	null	null	null	null	null	null	null	null	null
3	null	null	null	null	null	null	null	null	null
4	null	null	null	null	null	null	null	null	null
5	null	null	null	null	null	null	null	null	null
6	null	null	null	null	null	null	B	W	null

An additional call would result in the following change

```
move("B", 7)
```

	0	1	2	3	4	5	6	7	8
0	null	null	null	null	null	null	null	null	null
1	null	null	null	null	null	null	null	null	null
2	null	null	null	null	null	null	null	null	null
3	null	null	null	null	null	null	null	null	null
4	null	null	null	null	null	null	null	null	null
5	null	null	null	null	null	null	null	B	null
6	null	null	null	null	null	null	B	W	null

```
public boolean move(color, column)
```

Begin your response at the top of a new page in the separate Free Response booklet and fill in the appropriate circle at the top of each page to indicate the question number. If there are multiple parts to this question, write the part letter with your response.

GO ON TO THE NEXT PAGE.

(c) The method `checkVerticalWin(color)` will check all the columns to determine if there are four adjacent checkers of the specified color in any column. If so, the method will return the column containing those checkers. If four consecutive checkers of the specified color are not detected, the method will return –1.

`checkVerticalWin("W")` returns 7.

	0	1	2	3	4	5	6	7	8
0	null	null	null	null	null	null	null	null	null
1	null	null	null	null	null	null	null	null	null
2	null	null	null	null	null	null	null	null	null
3	null	null	null	B	null	W	null	W	null
4	null	null	null	W	B	B	null	W	null
5	null	null	null	W	B	B	null	W	null
6	null	null	null	W	B	B	B	W	null

```
public int checkVerticalWin(color)
```

Begin your response at the top of a new page in the separate Free Response booklet and fill in the appropriate circle at the top of each page to indicate the question number. If there are multiple parts to this question, write the part letter with your response.

STOP

END OF EXAM

The Princeton Review®

Completely darken bubbles with a No. 2 pencil. If you make a mistake, be sure to erase mark completely. Erase all stray marks.

1.

YOUR NAME: (Print) _____ Last _____ First _____ M.I.

SIGNATURE: _____ DATE: ___/___/___

HOME ADDRESS: (Print) _____ Number and Street

_____ City _____ State _____ Zip Code

PHONE NO.: _____

IMPORTANT: Please fill in these boxes exactly as shown on the back cover of your test book.

2. TEST FORM

3. TEST CODE

4. REGISTRATION NUMBER

0	A	J	0	0	0	0	0	0	0	0	0
1	B	K	1	1	1	1	1	1	1	1	1
2	C	L	2	2	2	2	2	2	2	2	2
3	D	M	3	3	3	3	3	3	3	3	3
4	E	N	4	4	4	4	4	4	4	4	4
5	F	O	5	5	5	5	5	5	5	5	5
6	G	P	6	6	6	6	6	6	6	6	6
7	H	Q	7	7	7	7	7	7	7	7	7
8	I	R	8	8	8	8	8	8	8	8	8
9			9	9	9	9	9	9	9	9	9

5. YOUR NAME

First 4 letters of last name					FIRST INIT	MID INIT
A	A	A	A		A	A
B	B	B	B		B	B
C	C	C	C		C	C
D	D	D	D		D	D
E	E	E	E		E	E
F	F	F	F		F	F
G	G	G	G		G	G
H	H	H	H		H	H
I	I	I	I		I	I
J	J	J	J		J	J
K	K	K	K		K	K
L	L	L	L		L	L
M	M	M	M		M	M
N	N	N	N		N	N
O	O	O	O		O	O
P	P	P	P		P	P
Q	Q	Q	Q		Q	Q
R	R	R	R		R	R
S	S	S	S		S	S
T	T	T	T		T	T
U	U	U	U		U	U
V	V	V	V		V	V
W	W	W	W		W	W
X	X	X	X		X	X
Y	Y	Y	Y		Y	Y
Z	Z	Z	Z		Z	Z

6. DATE OF BIRTH

Month	Day		Year	
JAN				
FEB	0	0	0	0
MAR	1	1	1	1
APR	2	2	2	2
MAY	3	3	3	3
JUN		4	4	4
JUL		5	5	5
AUG		6	6	6
SEP		7	7	7
OCT		8	8	8
NOV		9	9	9
DEC				

The Princeton Review®

1. A B C D E	11. A B C D E	21. A B C D E	31. A B C D E
2. A B C D E	12. A B C D E	22. A B C D E	32. A B C D E
3. A B C D E	13. A B C D E	23. A B C D E	33. A B C D E
4. A B C D E	14. A B C D E	24. A B C D E	34. A B C D E
5. A B C D E	15. A B C D E	25. A B C D E	35. A B C D E
6. A B C D E	16. A B C D E	26. A B C D E	36. A B C D E
7. A B C D E	17. A B C D E	27. A B C D E	37. A B C D E
8. A B C D E	18. A B C D E	28. A B C D E	38. A B C D E
9. A B C D E	19. A B C D E	29. A B C D E	39. A B C D E
10. A B C D E	20. A B C D E	30. A B C D E	40. A B C D E

Practice Test 3

AP® Computer Science A Exam

SECTION I: Multiple-Choice Questions

DO NOT OPEN THIS BOOKLET UNTIL YOU ARE TOLD TO DO SO.

At a Glance

Total Time
1 hour 30 minutes
Number of Questions
40
Percent of Total Score
50%
Writing Instrument
Pencil required

Instructions

Section I of this examination contains 40 multiple-choice questions. Fill in only the ovals for numbers 1 through 40 on your answer sheet.

Indicate all of your answers to the multiple-choice questions on the answer sheet. No credit will be given for anything written in this exam booklet, but you may use the booklet for notes or scratch work. After you have decided which of the suggested answers is best, completely fill in the corresponding oval on the answer sheet. Give only one answer to each question. If you change an answer, be sure that the previous mark is erased completely. Here is a sample question and answer.

Sample Question

Chicago is a
(A) state
(B) city
(C) country
(D) continent
(E) county

Sample Answer

Use your time effectively, working as quickly as you can without losing accuracy. Do not spend too much time on any one question. Go on to other questions and come back to the ones you have not answered if you have time. It is not expected that everyone will know the answers to all the multiple-choice questions.

About Guessing

Many candidates wonder whether or not to guess the answers to questions about which they are not certain. Multiple-choice scores are based on the number of questions answered correctly. Points are not deducted for incorrect answers, and no points are awarded for unanswered questions. Because points are not deducted for incorrect answers, you are encouraged to answer all multiple-choice questions. On any questions you do not know the answer to, you should eliminate as many choices as you can, and then select the best answer among the remaining choices.

GO ON TO THE NEXT PAGE.

Java Quick Reference

Class Constructors and Methods	Explanation
String Class	
`String(String str)`	Constructs a new `String` object that represents the same sequence of characters as `str`
`int length()`	Returns the number of characters in a `String` object
`String substring(int from, int to)`	Returns the substring beginning at index `from` and ending at index `to - 1`
`String substring(int from)`	Returns `substring(from, length())`
`int indexOf(String str)`	Returns the index of the first occurrence of `str`; returns `-1` if not found
`boolean equals(String other)`	Returns `true` if `this` is equal to `other`; returns `false` otherwise
`int compareTo(String other)`	Returns a value <0 if `this` is less than `other`; returns zero if `this` is equal to `other`; returns a value of >0 if `this` is greater than `other`
Integer Class	
`Integer(int value)`	Constructs a new `Integer` object that represents the specified `int` value
`Integer.MIN_VALUE`	The minimum value represented by an `int` or `Integer`
`Integer.MAX_VALUE`	The maximum value represented by an `int` or `Integer`
`int intValue()`	Returns the value of this `Integer` as an `int`
Double Class	
`Double(double value)`	Constructs a new `Double` object that represents the specified `double` value
`double doubleValue()`	Returns the value of this `Double` as a `double`
Math Class	
`static int abs(int x)`	Returns the absolute value of an `int` value
`static double abs(double x)`	Returns the absolute value of a `double` value
`static double pow(double base, double exponent)`	Returns the value of the first parameter raised to the power of the second parameter
`static double sqrt(double x)`	Returns the positive square root of a `double` value
`static double random()`	Returns a `double` value greater than or equal to `0.0` and less than `1.0`
ArrayList Class	
`int size()`	Returns the number of elements in the list
`boolean add(E obj)`	Appends `obj` to end of list; returns `true`
`void add(int index, E obj)`	Inserts `obj` at position index (`0 <= index <= size`), moving elements at position `index` and higher to the right (adds `1` to their indices) and adds 1 to size
`E get(int index)`	Returns the element at position `index` in the list
`E set(int index, E obj)`	Replaces the element at position `index` with `obj`; returns the element formerly at position `index`
`E remove(int index)`	Removes the element at position `index`, moving elements at position `index + 1` and higher to the left (subtracts `1` from their indices) and subtracts 1 from size; returns the element formerly at position `index`
Object Class	
`boolean equals(Object other)`	
`String toString()`	

GO ON TO THE NEXT PAGE.

COMPUTER SCIENCE A

SECTION I

Time—1 hour and 30 minutes

Number of Questions—40

Percent of total exam grade—50%

Directions: Determine the answer to each of the following questions or incomplete statements, using the available space for any necessary scratchwork. Then decide which is the best of the choices given and fill in the corresponding oval on the answer sheet. No credit will be given for anything written in the examination booklet. Do not spend too much time on any one problem.

Notes:

- Assume that the classes listed in the Java Quick Reference have been imported where appropriate.
- Assume that declarations of variables and methods appear within the context of an enclosing class.
- Assume that method calls that are not prefixed with an object or class name and are not shown within a complete class definition appear within the context of an enclosing class.
- Unless otherwise noted in the question, assume that parameters in the method calls are not `null` and that methods are called only when their preconditions are satisfied.

1. Consider the following methods.

```
public void trial()
{
      int a = 10;
      int b = 5;
      doubleValues(a, b);
      System.out.print(b);
      System.out.print(a);
}

public void doubleValues(int c, int d)
{
      c = c * 2;
      d = d * 2;
      System.out.print(c);
      System.out.print(d);
}
```

What is printed as the result of the call `trial()`?

(A) `2010`
(B) `2010105`
(C) `2010510`
(D) `20102010`
(E) `20101020`

GO ON TO THE NEXT PAGE.

2. Consider the following method.

```
/**
 * Precondition: a > b > 0
 */
public static int mystery(int a, int b)
{
   int d = 0;
   for (int c = a; c > b; c--)
   {
      d = d + c;
   }
   return d;
}
```

What is returned by the call `mystery(x, y)`?

(A) The sum of all integers greater than `y` but less than or equal to `x`
(B) The sum of all integers greater than or equal to `y` but less than or equal to `x`
(C) The sum of all integers greater than `y` but less than `x`
(D) The sum of all integers greater than or equal to `y` but less than `x`
(E) The sum of all integers less than `y` but greater than or equal to `x`

3. Consider the following method.

```
public void mystery (int  n)
{
   int k;
   for (k = 0 ; k < n ; k++)
   {
      mystery(k);
      System.out.print (k);
   }
}
```

What is printed by the call `mystery(3)` ?

(A) `0123`
(B) `00123`
(C) `0010012`
(D) `00100123`
(E) `001001200100123`

GO ON TO THE NEXT PAGE.

4. Consider an array of integers.

```
4    10    1    2    6    7    3    5
```

If selection sort is used to order the array from smallest to largest values, which of the following represents a possible state of the array at some point during the selection sort process?

(A) `1  4  10  2  3  6  7  5`
(B) `1  2  4  6  10  7  3  5`
(C) `1  2  3  10  6  7  4  5`
(D) `4  3  1  2  6  7  10  5`
(E) `5  3  7  6  2  1  10  4`

5. Consider the following code segment:

```
int  k;
int  a[];
a = new int [7];
for (k = 0; k < a.length; k++)
{
   a[k] = a.length - k;
}
for (k = 0; k < a.length - 1; k++)
{
   a[k+1] = a[k];
}
```

What values will A contain after the code segment is executed?

(A) `1  1  2  3  4  5  6`
(B) `1  2  3  4  5  6  7`
(C) `6  6  5  4  3  2  1`
(D) `7  7  6  5  4  3  2`
(E) `7  7  7  7  7  7  7`

GO ON TO THE NEXT PAGE.

Questions 6–7 refer to the following two classes.

```
public class PostOffice
{
    // constructor initializes boxes
    // to length 100
    public PostOffice()
    {   /* implementation not shown   */}

    // returns the P.O. Box based on the given P.O. Box number
    //  0 <=  theBox  <  getNumBoxes ()
    public Box getBox(int theBox)
    {   /* implementation not shown   */}
    // returns the number of p.o. boxes
    public int getNumBoxes()
    {   /* implementation not shown   */}

    // private data members and
    // other methods not shown
}

public class Box
{
    // constructor
    public Box()
    {   /* implementation not shown  */}

    // returns the number of this box
    public int getBoxNumber()
    {   /* implementation not shown   */}

    // returns the number of pieces
    // of mail in this box
    public int getMailCount()
    {   /* implementation not shown  */}
    // returns the given piece of mail
    // 0 <= thePiece < getMailCount ( )
    public Mail getMail(int thePiece)
    {   /* implementation not shown  */}
    // true if the box has been assigned
    // to a customer
    public boolean isAssigned()
    {   /* implementation not shown  */}
    // true if the box contains mail
    public boolean hasMail()
    {   /* implementation not shown  */}
    // private data members and
    // other methods not shown
}
public class Mail
{
    // private members,  constructors, and
    // other methods not shown
}
```

GO ON TO THE NEXT PAGE.

6. Consider the following code segment:

```
PostOffice p[ ];
p = new PostOffice[10];
```

Assuming that the box has been assigned and that it has at least four pieces of mail waiting in it, what is the correct way of getting the fourth piece of mail from the 57th box of the 10th post office of `p`?

(A) `Mail m = p[10].getBox(57).getmail(4);`
(B) `Mail m = p[9].getBox(56).getMail(3);`
(C) `Mail m = p.getMail(57).getMail(4) [10];`
(D) `Mail m = getMail(getBox(p[9], 560, 3);`
(E) `Mail m = new Mail(10, 57, 4);`

GO ON TO THE NEXT PAGE.

7. Consider the incomplete function `printEmptyBoxes` given below. `printEmptyBoxes` should print the box numbers of all of the boxes that have been assigned to a customer but do not contain mail.

```
public void printEmptyBoxes (PostOffice[] p)
{
   for (int k = 0; k < p.length - 1 ; k++)
   {
      for (int x = 0; x < p[k].getNumBoxes() - 1 ; x++)
      {
           /* missing code */
      }
   }
}
```

Which of the following could be used to replace `/* missing code */` so that `printBoxesWithoutMail` works as intended?

(A)
```
if (p[k].getBox(x).isAssigned() &&
!p[k].getBox(x).hasMail())
{
     System.out.println(p[k].getBox(x).getBoxNumber()) ;
}
```

(B)
```
if (p[x].getBox(k).isAssigned() &&
!p[x].getBox(k).hasMail())
{
  System.out.println(p[x].getBox(k).getBoxNumber());
}
```

(C)
```
if (p[k].getBox(x).isAssigned() &&
!p[k].getBox(x).hasMail())
{
     System.out.println (p[k].getBoxNumber (x));
}
```

(D)
```
if (p[x].getBox(k).isAssigned() &&
!p[x].getBox (k).hasMail())
{
     System.out.println(p[x].getBoxNumber(k));
  }
```

(E)
```
if (p[x].getBox(k).isAssigned() &&
p[x].getBox(k).getMail() == 0)
{
     System.out.println(k);
}
```

GO ON TO THE NEXT PAGE.

8. Assume that a and b are boolean variables that have been initialized. Consider the following code segment.

```
a = a && b;
b = a || b;
```

Which of the following statements is always true?

I. The final value of a is the same as the initial value of a.
II. The final value of b is the same as the initial value of b.
III. The final value of a is the same as the initial value of b.

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

9. Consider the following code segment.

```
int x;
x = 53;
if (x > 10)
{
   System.out.print("A");
}
if (x > 30)
{
   System.out.print("B");
}
else if (x > 40)
{
   System.out.print("C");
}
if (x > 50)
{
   System.out.print ("D");
}
if (x > 70)
{
   System.out.print ("E");
}
```

What is the output when the code is executed?

(A) A
(B) D
(C) ABD
(D) ABCD
(E) ABCDE

GO ON TO THE NEXT PAGE.

10. Consider the following code segment:

```
int j;
int k;
for (j = -2; j <= 2; j = j + 2)
{
   for (k = j; k < j + 3; k++)
   {
      System.out.print(k + " ");
   }
}
```

What is the output when the code is executed?

(A) -2 -1 0
(B) -2 -1 0 1 2
(C) 0 1 2 0 1 2 0 1 2
(D) -2 0 2
(E) -2 -1 0 0 1 2 2 3 4

11. Consider the following method.

```
public void mystery (int  count, String s)
{
   if  (count  <=  0)
   {
      return;
   }
   if  (count % 3 == 0)
   {
      System.out.print(s + "--" + s);
   }
   else if (count % 3 == 1)
   {
      System.out.print(s + "-" + s);
   }
   else
   {
      System.out.print(s);
   }
   mystery(count - 1, s);
}
```

What is outputted by the call `mystery(5, "X")`?

(A) XX-XX--XXX-X
(B) XX-XX-XX-XX
(C) XXX--XX-X-XX--XXX
(D) XX-XXX--XXX-XX
(E) XXXXX

GO ON TO THE NEXT PAGE.

Questions 12–13 refer to the following classes and method descriptions.

Class `Table` has a method, `getPrice`, which takes no parameters and returns the price of the table.

Class `Chair` also has a method, `getPrice`, which takes no parameters and returns the price of the chair.

Class `DiningRoomSet` has a constructor which is passed a `Table` object and an `ArrayList` of `Chair` objects. It stores these parameters in its private data fields `myTable` and `myChairs`.

Class `DiningRoomSet` has a method, `getPrice`, which takes no parameters and returns the price of the dining room set. The price of a dining room set is calculated as the sum of the price of its table and all of its chairs.

12. What is the correct way to define the signature of the constructor for the `DiningRoomSet` class?

(A) `public void DiningRoomSet(Table t, ArrayList, chairs)`
(B) `public DiningRoomSet(Table t, ArrayList<Chair> chairs)`
(C) `public void DiningRoomSet(Table t, ArrayList Chair Chairs)`
(D) `public DiningRoomSet(Table t, ArrayList Chair Chairs)`
(E) `public DiningRoomSet(Table t, Chair Chairs)`

13. What is the correct way to implement the getPrice method of the `DiningRoomSet` class?

(A)
```
public double getPrice(Table t, ArrayList chairs)
{
     return t.getPrice() + chairs.getPrice();
}
```
(B)
```
public double getPrice(Table t, ArrayList chairs)
{
     return myTable.getPrice() + myChairs.getPrice();
}
```
(C)
```
public double getPrice()
{
     return myTable.getPrice() + myChairs.getPrice();
}
```
(D)
```
public double getPrice()
{
     double result  = myTable.getPrice();
     for (int k = 0; k < myChairs.size() - 1; k++)
     {
          result += ((Chair)myChairs.get(k)).getPrice();
     }
     return result;
}
```
(E)
```
public double getPrice()
{
      double result = myTable.getPrice();
      for (int k = 0; k  <  myChairs.length - 1; k++)
      {
           result += ((Chair)myChairs[k]).getPrice();
      }
      return result;
}
```

GO ON TO THE NEXT PAGE.

14. Consider the following output:

```
6   5   4   3   2   1
5   4   3   2   1
4   3   2   1
3   2   1
2   1
1
```

Which of the following code segments produces the above output when executed?

(A)
```
for  (int j = 6;  j  <  0;  j--)
{
  for (int k = j; k > 0; k--)
  {
      System.out.print(k  +  "   ");
  }
  System.out.println(" ");
}
```

(B)
```
for (int j = 6; j >= 0; j--)
{
  for (int k = j; k >= 0; k--)
  {
      System.out.print(k + " ");
  }
  System.out.println(" ");
}
```

(C)
```
for (int j = 0; j < 6; j++)
{
  for (int k = 6 - j; k > 0; k--)
  {
      System.out.print(k + "  ");
  }
  system.out.println(" ");
}
```

(D)
```
for (int j = 0; j < 6; j++)
{
  for (int k = 7 - j ; k > 0 ; k--)
  {
      System.out.print(k + " ");
  }
  System.out.println(" ");
}
```

(E)
```
for (int j = 0; j < 6; j++)
{
  for (int k = 6  -  j ; k >= 0; k--)
  {
      System.out.print(k + " ");
  }
  System.out.println(" ");
}
```

GO ON TO THE NEXT PAGE.

15. Consider the following code segment.

```
List<Integer> list = new ArrayList<Integer>();
list.add(new Integer(7));
list.add(new Integer(6));
list.add(1, new Integer(5));
list.add(1, new Integer(4));
list.add(new Integer(3));
list.set(2, new Integer(2));
list.add(1, new Integer(1));
System.out.println(list);
```

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

(A) `[1, 4, 2, 7, 6, 3]`
(B) `[7, 1, 4, 2, 6, 3]`
(C) `[7, 2, 5, 4, 3, 1]`
(D) `[7, 6, 2, 4, 3, 1]`
(E) `[7, 1, 2]`

16. Consider the following declarations.

```
public class Animal
{
    String makeSound()
    {
       // Implementation not shown
    }
    String animalType()
    {
       // Implementation not shown
    }
}
public static class Dog extends Animal
{
    public String makeSound(Animal a)
    {
       // Implementation not shown
    }
}
```

Which of the following methods must be included in the declaration of the `Dog` class in order for the class to successfully compile?

I. `public String makeSound()`
II. `public String animalType()`
III. `public String animalType(Animal b)`

(A) I only
(B) II only
(C) I and II only
(D) II and III only
(E) None

17. Consider the following two classes.

```
public class Fish
{
  public String endoskeleton = “bone”;

  public void action()
  {
      System.out.println(“splash splash”);
  }
}

public class Shark extends Fish
{
  public void action()
  {
      System.out.println(“chomp chomp”);
  }

  public String endoskeleton = “cartilage”;
}
```

Which of the following is the correct output after the following code segment is executed?

```
Fish Bob = new Shark();
System.out.println(Bob.endoskeleton);
Bob.action();
```

(A) bone
chomp chomp

(B) bone
splash splash

(C) cartilage
splash splash

(D) cartilage
chomp chomp

(E) cartilage
splash splash
chomp chomp

GO ON TO THE NEXT PAGE.

Questions 18–19 refer to the following incomplete method.

The following `insertSort` method sorts the values in an integer array, `sort`, in ascending order.

```
1   public static void insertSort(int[] sort)
2      {
3         for (int index = 1;index < sort.length;index++)
4         {
5         int temp = sort[index];
6         while (index > 0 && sort[index - 1] > temp)
7               {
8         /* missing code */
9               }
10     sort[index] = temp;
11        }
12     }
```

18. Which of the following can be used to replace `/* missing code */` so that the `insertSort` method will execute properly?

(A)
```
sort[index] = sort[index - 1];
index++;
```
(B)
```
sort[index - 1] = sort[index];
index--;
```
(C)
```
sort[index] = sort[index + 1];
index++;
```
(D)
```
sort[index] = sort[index - 1];
index--;
```
(E)
```
sort[index] = sort[index + 1];
index--;
```

19. Assuming that the `/* missing code */` is implemented properly, what change can be made to the code in order for the array to be sorted in descending order?

(A) Replace Line 6 with: `while (index < 0 && sort[index - 1] > temp)`
(B) Replace Line 6 with: `while (index < 0 && sort[index - 1] < temp)`
(C) Replace Line 6 with: `while (index > 0 && sort[index - 1] < temp)`
(D) Replace Line 3 with: `for (int index = sort.length - 1;index > 0;index--)`
(E) Replace Line 3 with: `for (int index = 1;index > 0;index--)`

20. Which of the following arrays would be sorted the slowest using insertion sort?

(A) `[3 4 6 2 7 3 9]`
(B) `[3 2 5 4 6 7 9]`
(C) `[9 7 6 5 4 3 2]`
(D) `[2 3 4 5 6 7 9]`
(E) `[9 3 2 4 5 7 6]`

GO ON TO THE NEXT PAGE.

Questions 21–23 refer to the following incomplete class declaration used to represent fractions with integer numerators and denominators.

```
public class Fraction
{
  private int numerator;
  private int denominator;

  public Fraction()
  {
      numerator = 0;
      denominator = 1;
  }

  public Fraction(int n, int d)
  {
      numerator = n;
      denominator = d;
  }

  // postcondition: returns the
  //     numerator
  public int getNumerator()
  {   /* implementation not shown */  }

  // postcondition: returns the
  //     denominator
  public int getDenominator()
  {   /* implementation not shown*/   }

  // postcondition: returns the greatest
  // common divisor of x and y
  public int gcd(int x, int y)
  {   /* implementation not shown*/   }

  // postcondition: returns the Fraction
  //     that is the result of multiplying
  //     this Fraction and f

  public Fraction multiply(Fraction f)
  {   /* implementation not shown */  }
  // . . . other methods not shown
}
```

GO ON TO THE NEXT PAGE.

21. Consider the method `multiply` of the `Fraction` class.

```
// postcondition: returns the Fraction
//          that is the result of multiplying
//          this Fraction and f
public Fraction multiply(Fraction f)
{  /* missing code */  }
```

Which of the following statements can be used to replace `/* missing code */` so that the `multiply` method is correctly implemented?

I.
```
return Fraction(
  numerator * f.getNumerator(),
  denominator * f.getDenominator());
```

II.
```
return new Fraction(
  numerator * f.numerator,
  denominator * f.denominator());
```

III.
```
return new Fraction(
  numerator * f.getNumerator(),
  denominator * f.getDenominator());
```

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

22. Consider the use of the `Fraction` class to multiply the fractions $\frac{3}{4}$ and $\frac{7}{19}$. Consider the following code:

```
Fraction fractionOne;
Fraction fractionTwo;
Fraction answer;
fractionOne = new Fraction(3, 4);
fractionTwo = new Fraction(7, 19);
/* missing code */
```

Which of the following could be used to replace `/* missing code */` so that the answer contains the result of multiplying fractionOne by fractionTwo?

(A)
```
answer = fractionOne * fractionTwo
```
(B)
```
answer = multiply(fractionOne,fractionTwo);
```
(C)
```
answer = fractionOne.multiply(fractionTwo);
```
(D)
```
answer = new Fraction(fractionOne,fractionTwo);
```
(E)
```
answer = (fractionOne.getNumerator() * fractionTwo.getNumerator()) /
      (fractionOne.getDenominator() * fractionTwo.getDenominator());
```

GO ON TO THE NEXT PAGE.

23. The following incomplete class declaration is intended to extend the `Fraction` class so that fractions can be manipulated in reduced form (lowest terms).

Note that a fraction can be reduced to lowest terms by dividing both the numerator and denominator by the greatest common divisor (gcd) of the numerator and denominator.

```
public class ReducedFraction extends Fraction
{
   private int reducedNumerator;
   private int reducedDenominator;
   // . . . constructors and other methods not shown
}
```

Consider the following proposed constructors for the `ReducedFraction` class:

I.
```
public ReducedFraction( )
{
  reducedNumerator = 0;
  reducedDenominator = 1;
}
```

II.
```
public ReducedFraction(int n, int d)
{
  numerator = n;
  denominator = d;
  reducedNumerator = n / gcd(n, d);
  reducedDenominator = d / gcd(n, d);
}
```

III.
```
public ReducedFraction(int n, int d)
{
  super(n, d);
  reducedNumerator = n / gcd(n, d);
  reducedDenominator = d / gcd(n, d);
}
```

Which of these constructor(s) would be legal for the `ReducedFraction` class?

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

GO ON TO THE NEXT PAGE.

24. Consider `s1` and `s2` defined as follows.

```
String s1 = new String("hello");
String s2 = new String("hello");
```

Which of the following is/are correct ways to see if `s1` and `s2` hold identical strings?

I.
```
if (s1 == s2)
  /* s1 and s2 are identical */
```
II.
```
if (s1.equals(s2))
  /* s1 and s2 are identical */
```
III.
```
if (s1.compareTo(s2) == 0)
  /* s1 and s2 are identical */
```

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

25. Consider the following variable and method declarations:

```
String s;
String t;
public void mystery (String a, String b)
{
   a = a + b;
   b = b + a;
}
```

Assume that `s` has the value "`Elizabeth`" and `t` has the value "`Andrew`" and `mystery(s, t)` is called. What are the values of `s` and `t` after the call to `mystery`?

	`s`	`t`
(A)	Elizabeth	Andrew
(B)	ElizabethAndrew	AndrewElizabeth
(C)	ElizabethAndrew	AndrewElizabethAndrew
(D)	ElizabethAndrew	ElizabethAndrewAndrew
(E)	ElizabethAndrewElizabeth	AndrewElizabethAndrew

GO ON TO THE NEXT PAGE.

26. Consider the following incomplete and *incorrect* class and interface declarations:

```
public class Comparable Object
{
   public int compareTo(Object o)
   {
      //method body not shown
   }
//other methods and variables not shown
}
public class Point extends ComparableObject
{
   private int x;
   private int y;
   public boolean compareTo(Point other)
   {
      return (x == other.x &&
              y == other.y);
   }
   // . . . constructors and other methods
   //    not shown
}
```

For which of the following reasons is the above class structure incorrect?

I. Objects may not access private data fields of other objects in the same class.
II. The `ComparableObject` class requires that `compareTo` be passed as an `Object` rather than a `Point`.
III. The `ComparableObject` class requires that `compareTo` return an `int` rather than a `boolean`.

(A) I only
(B) III only
(C) I and III only
(D) II and III only
(E) None, the above class declarations are correct.

GO ON TO THE NEXT PAGE.

27. Consider the following abstraction of a `for` loop where `<1>, <2>, <3>,` and `<4>` represent legal code in the indicated locations:

```
for (<1>; <2>; <3>)
{
   <4>
}
```

Which of the following `while` loops has the same functionality as the above `for` loop?

(A)
```
<1>;
while (<2>)
{
   <3>;
   <4>
}
```

(B)
```
<1>;
while (<2>)
{
   <4>
   <3>;
}
```

(C)
```
<1>;
   while (!<2>)
{
   <3>;
   <4>
}
```

(D)
```
<1>;
while (!<2>)
{
   <4>
   <3>;
}
```

(E)
```
<1>;
<3>;
while (<2>)
{
   <4>
   <3>;
}
```

28. Consider the following expression:

```
a / b + c - d % e * f
```

Which of the expressions given below is equivalent to the one given above?

(A) `((a / b) + (c - d)) % (e * f)`
(B) `((((a / b) + c) - d) % e) * f`
(C) `((a / b) + c) - (d % (e * f)`
(D) `(a / ((b + c) - d) % e) * f`
(E) `((a / b) + c) - ((d % e) * f)`

GO ON TO THE NEXT PAGE.

29. Assume that a program declares and initializes x as follows:

```
String[] x ;
x = new String[10] ;
initialize(x);          // Fills the array x with
                        // valid strings each of
                        // length 5
```

Which of the following code segments correctly traverses the array and prints out the first character of all ten strings followed by the second character of all ten strings, and so on?

I.
```
int i;
int j;
for (i = 0 ; i < 10 ; i++)
     for (j = 0 ; j < 5 ; j++)
         System.out.print(x[i].substring(j, j + 1));
```

II.
```
int i;
int j;
for (i = 0 ; i < 5 ; i++)
    for (j = 0 ; j < 10 ; j++)
        System.out.print(x[j].substring(i, i + 1));
```

III.
```
int i ;
int j ;
for (i = 0 ; i < 5 ; i++)
    for (j = 0 ; j < 10 ; j++)
        System.out.print(x[i].substring(j, j + 1));
```

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

GO ON TO THE NEXT PAGE.

30. Consider the following declaration and assignment statements:

```
int a = 7;
int b = 4;
double c;
c = a / b;
```

After the assignment statement is executed, what's the value of `c`?

(A) 1.0
(B) 1.75
(C) 2.0
(D) An error occurs because `c` was not initialized.
(E) An error occurs because `a` and `b` are integers and `c` is a double.

31. Consider the following code segment:

```
int x;
x = /* initialized to an integer */
if (x % 2 == 0 && x / 3 == 1)
   System.out.print("Yes");
```

For what values of `x` will the word "Yes" be printed when the code segment is executed?

(A) 0
(B) 4
(C) Whenever `x` is even and `x` is not divisible by 3
(D) Whenever `x` is odd and `x` is divisible by 3
(E) Whenever `x` is even and `x` is divisible by 3

GO ON TO THE NEXT PAGE.

32. Consider the following incomplete class definition:

```
public class SomeClass
{
   private String myName;
   // postcondition: returns myName
   public String getName( )
   { /* implementation not shown */ }
   // postcondition: myName == name
   public void setName(String name)
   { /* implementation not shown */ }
   // . . . constructors, other methods
   //    and private data not shown
}
```

Now consider the method `swap`, not part of the `SomeClass` class.

```
// precondition: x and y are correctly
//    constructed
// postcondition: the names of objects
//    x and y are swapped
public void swap (SomeClass x, SomeClass y)
{
   /* missing code */
}
```

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

I.
```
SomeClass temp;
temp = x;
x = y;
y = temp;
```

II.
```
String temp;
temp = x.myName;
x.myName = y.myName;
y.myName = temp;
```

III.
```
String temp;
temp = x.getName();
x.setName(y.getName());
y.setName(temp);
```

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

GO ON TO THE NEXT PAGE.

33. A bookstore wants to store information about the different types of books it sells.

For each book, it wants to keep track of the title of the book, the author of the book, and whether the book is a work of fiction or nonfiction.

If the book is a work of fiction, then the bookstore wants to keep track of whether it is a romance novel, a mystery novel, or science fiction.

If the book is a work of nonfiction, then the bookstore wants to keep track of whether it is a biography, a cookbook, or a self-help book.

Which of the following is the best design?

(A) Use one class, `Book`, which has three data fields: `String title`, `String author`, and `int bookType`.
(B) Use four unrelated classes: `Book`, `Title`, `Author`, and `BookType`.
(C) Use a class `Book` which has two data fields: `String title`, `String author`, and a subclass: `BookType`.
(D) Use a class `Book` which has two data fields: `String title`, `String author`, and six subclasses: `RomanceNovel`, `Mystery`, `ScienceFiction`, `Biography`, `Cookbook`, and `SelfHelpBook`.
(E) Use a class `Book` which has two data fields: `String title`, `String author`, and two subclasses: `FictionWork` and `NonFictionWork`. The class `FictionWork` has three subclasses, `RomanceNovel`, `Mystery`, and `ScienceFiction`. The class `NonFictionWork` has three subclasses: `Biography`, `Cookbook`, and `SelfHelpBook`.

GO ON TO THE NEXT PAGE.

34. Consider the following code:

```
public int mystery(int x)
{
   if (x == 1)
      return <missing value>;
   else
      return(2 * mystery(x - 1)) + x;
}
```

Which of the following can be used to replace `<missing value>` so that `mystery (4)` returns `34` ?

(A) 0
(B) 1
(C) 2
(D) 3
(E) 4

35. Consider the following code segment:

```
int [ ] X;
int [ ] Y;
int k;
X = initializeX();       // returns a valid
                         // initialized int [ ]
Y = initializeY();       // returns a valid
                         // initialized int [ ]
for (k = 0;
   k < X.length && X[k] == Y[k];
   k++)
{
   /* some code */
}
```

Assuming that after `X` and `Y` are initialized, `X.length == Y.length`, which of the following must be true after executing this code segment?

(A) `k < X.length`
(B) `k < X.length && X[k] == Y[k]`
(C) `k < X.length && X[k] != Y[k]`
(D) `k >= X.length || X[k] == Y[k]`
(E) `k >= X.length || X[k] != Y[k]`

36. Which of the following would NOT cause a run-time exception?

(A) Dividing an integer by zero
(B) Using an object that has been declared but not instantiated
(C) Accessing an array element with an array index that is equal to the length of the array
(D) Attempting to create a substring beginning at a negative index
(E) Attempting to call a method with the wrong number of arguments

GO ON TO THE NEXT PAGE.

37. Assume that a and b are properly initialized variables of type `Double`.

Which of the following is an equivalent expression to:

```
a.doubleValue()   != b.doubleValue()
```

(A) `a != b`
(B) `a.notEquals(b)`
(C) `!(a.doubleValue() .equals(b.doubleValue()))`
(D) `!(a.compareTo(b))`
(E) `a.compareTo(b) != 0`

38. Which of the following would be the LEAST effective way of ensuring reliability in a program?

(A) Encapsulating functionality in a class by declaring all data fields to be public
(B) Defining and following preconditions and postconditions for every method
(C) Including assertions at key places in the code
(D) Using descriptive variable names
(E) Indenting code in a consistent and logical manner

39. Consider a dictionary that has 1,024 pages with 50 words on each page.

In order to look up a given target word, a student is considering using one of the following three methods:

Method 1

Use a binary search technique to find the correct page (comparing the target word with the first word on a given page). When the correct page is found, use a sequential search technique to find the target word on the page.

Method 2

Use a sequential search technique to find the correct page (comparing the target word with the first word on a given page). When the correct page is found, use another sequential search technique to find the target word on the page.

Method 3

Use a sequential search technique on all of the words in the dictionary to find the target word.

Which of the following best characterizes the greatest number of words that will be examined using each method?

	Method 1	Method 2	Method 3
(A)	10	50	1,024
(B)	55	512	2,560
(C)	55	537	25,600
(D)	60	1,074	1,074
(E)	60	1,074	51,200

GO ON TO THE NEXT PAGE.

40. Consider the following recursive method.

```
public static int mystery(int m)
{
   if (m == 0)
   {
      return 0;
   }
   else
   {
      return 4 + mystery(m - 2);
   }
}
```

Assuming that j is a positive integer and that m = 2j, what value is returned as a result of the call `mystery(m)`?

(A) `0`
(B) `m`
(C) `2m`
(D) `j`
(E) `2j`

END OF SECTION I

IF YOU FINISH BEFORE TIME IS CALLED,
YOU MAY CHECK YOUR WORK ON THIS SECTION.

DO NOT GO ON TO SECTION II UNTIL YOU ARE TOLD TO DO SO.

GO ON TO THE NEXT PAGE.

COMPUTER SCIENCE A

SECTION II

Time—1 hour and 30 minutes

Number of Questions—4

Percent of Total Grade—50%

Directions: SHOW ALL YOUR WORK. REMEMBER THAT PROGRAM SEGMENTS ARE TO BE WRITTEN IN JAVA™.

Notes:

- Assume that the classes listed in the Java Quick Reference have been imported where appropriate.
- Unless otherwise noted in the question, assume that parameters in method calls are not `null` and that methods are called only when their preconditions are satisfied.
- In writing solutions for each question, you may use any of the accessible methods that are listed in classes defined in that question. Writing significant amounts of code that can be replaced by a call to one of these methods will not receive full credit.

FREE-RESPONSE QUESTIONS

1. A day care has a program to keep track of its employees and which children they teach during the day. An `Employee` has a minimum and maximum age they can teach. The `DayCare` also has a maximum ratio that specifies the maximum number of children a single employee can teach. Below is the full `DayCare` class:

```
public class DayCare
{
    private ArrayList<Employee> employees;
    private ArrayList<Child> children;
    private int maxRatio;

    public DayCare(int maxRatio)
    {
       employees = new ArrayList<Employee>();
       children = new ArrayList<Child>();
       this.maxRatio = maxRatio;
    }

    public boolean findEmployeeForChild(Child c)
    {
       /* To be completed in part (a) */
    }

    public boolean runDayCare()
    {
       /* To be completed in part (b) */
    }

    public boolean addChild(Child c)
    {
       /* To be completed in part (c) */
    }
}
```

GO ON TO THE NEXT PAGE.

(a) An `Employee` can only teach children between the employee's minimum age (inclusive) and maximum age (inclusive). They can also only teach children up to the day care's maximum ratio (inclusive). Below is the full `Employee` class.

```
public class Employee
{
    /* Instance variables not shown */

    public Employee(String name, String id, int min, int max)
    {
        /* Implementation not shown */
    }

    // Return the number of children currently assigned to this Employee
    public int childrenAssigned()
    {
        /* Implementation not shown */
    }

    // Assign a new child to this Employee
    public void assignChild(Child c)
    {
        /* Implementation not shown */
    }

    // Determine whether this Employee can teach a Child based on the child's age
    public boolean canTeach(int age)
    {
        /* Implementation not shown */
    }
}
```

A `Child` has accessors to get their name and age. While the implementation of `Child` is not shown, you can assume the accessors are called `getName` and `getAge`.

Complete the `findEmployeeForChild` method below that assigns a `Child` to the first `Employee` who can teach the `Child` and who has not reached the maximum ratio of the `DayCare`.

```
/* Return true if an Employee was found for the Child, false otherwise */
public boolean findEmployeeForChild(Child c)
```

Begin your response at the top of a new page in the separate Free Response booklet and fill in the appropriate circle at the top of each page to indicate the question number. If there are multiple parts to this question, write the part letter with your response.

GO ON TO THE NEXT PAGE.

(b) In order for the `DayCare` to run for a day, each `Child` must be assigned an `Employee`. If an `Employee` cannot be found for a `Child`, the `DayCare` cannot run for the day.

Complete the `runDayCare` method below that finds an `Employee` for each `Child` in the `children ArrayList`.

```
/* Return true if an Employee was found for each Child, false otherwise */
public boolean runDayCare()
```

Begin your response at the top of a new page in the separate Free Response booklet and fill in the appropriate circle at the top of each page to indicate the question number. If there are multiple parts to this question, write the part letter with your response.

GO ON TO THE NEXT PAGE.

(c) When a `Child` is added to the roster of the `DayCare`, the `DayCare` should first make sure there is an `Employee` available to teach that `Child`.

Complete the `addChild` method below that adds a `Child` to the `children ArrayList` if an `Employee` is available to teach that `Child`.

```
/* Return true if the Child was added to the ArrayList, false otherwise */
public boolean addChild(Child c)
```

Begin your response at the top of a new page in the separate Free Response booklet and fill in the appropriate circle at the top of each page to indicate the question number. If there are multiple parts to this question, write the part letter with your response.

GO ON TO THE NEXT PAGE.

2. A baseball team consists of different people including players, coaches, and people who work in the front office making trades and other transactions. The following `Person` class is used for all of the people who work for the team.

Each person has a name and an age.

```
public class Person
{
    private String fullName;
    private int age;

    public Person(String s, int a)
    {
       fullName = s;
       age = a;
    }

    // Accessors for name and age
    public String getName()
    {
       return fullName;
    }

    public int getAge()
    {
       return age;
    }
}
```

GO ON TO THE NEXT PAGE.

A `Player` has a name and age just like any person on the team, but also has a `position`. The position could be something like "catcher," "left fielder," or "infielder." Players should also be able to change their positions using a method called `changePosition`. Here is an example of a `Player` object:

```
Player p = new Player("Sammy Sosa", 32, "right fielder");
p.changePosition("outfielder");
```

Write the entire `Player` class.

Begin your response at the top of a new page in the separate Free Response booklet and fill in the appropriate circle at the top of each page to indicate the question number. If there are multiple parts to this question, write the part letter with your response.

GO ON TO THE NEXT PAGE.

3. A class is designed to store someone's full name. You can assume the name has only one space between the first name and the last name. The class has methods to extract the first name, last name, and number of vowels in the name. You can see an example below.

```
String fullName = "Katherine Johnson";
Name.getFirstName(fullName); // Returns "Katherine"
Name.getLastName(fullName); // Returns "Johnson"
Name.countVowels(fullName); // Returns 6
```

(a) The `getFirstName` method returns the first name based on a given full name. You can assume that `fullName` has only one space between the first name and the last name. Write the `getFirstName` method.

```
public static String getFirstName(String name)
```

Begin your response at the top of a new page in the separate Free Response booklet and fill in the appropriate circle at the top of each page to indicate the question number. If there are multiple parts to this question, write the part letter with your response.

GO ON TO THE NEXT PAGE.

(b) The `getLastName` method returns the last name based on a given full name. You can assume that `fullName` has only one space between the first name and the last name. Write the `getLastName` method.

```
public static String getLastName(String name)
```

Begin your response at the top of a new page in the separate Free Response booklet and fill in the appropriate circle at the top of each page to indicate the question number. If there are multiple parts to this question, write the part letter with your response.

GO ON TO THE NEXT PAGE.

(c) The `countVowels` method counts the number of vowels in the given full name. You can assume we will count only the letters `a`, `e`, `i`, `o`, and `u` as vowels. Write the entire `countVowels` method.

```
public static int countVowels(String name)
```

Begin your response at the top of a new page in the separate Free Response booklet and fill in the appropriate circle at the top of each page to indicate the question number. If there are multiple parts to this question, write the part letter with your response.

GO ON TO THE NEXT PAGE.

4. A city parking lot has a sign that keeps track of how many parking spaces are available in the lot. The class for the parking lot is detailed below.

```
public class ParkingLot
{
    private Car[ ][ ] lot;

    public ParkingLot(int rows, int cols)
    {
       lot = new Car[rows][cols];
    }

    public int openSpaces()
    {
       // Complete in part (a)
    }

    public boolean parkCar(Car newCar)
    {
       // Complete in part (b)
    }
}
```

GO ON TO THE NEXT PAGE.

(a) Write the `openSpaces` method that returns the number of spaces available in the parking lot. If a space is empty, it will be equal to `null`.

```
/* Return the number of empty spaces in the parking lot */
public int openSpaces( )
```

Begin your response at the top of a new page in the separate Free Response booklet and fill in the appropriate circle at the top of each page to indicate the question number. If there are multiple parts to this question, write the part letter with your response.

GO ON TO THE NEXT PAGE.

(b) Complete the `parkCar` method that puts a new car in any space in the parking lot and returns `true` if it was able to do so. It should return `false` if there are no empty spaces. You should use the `openSpaces` method to receive full credit.

```
/* Return true if there is an open spot to park the newCar, false otherwise. The car should be added to the lot 2D array
if there is an open spot. */
public boolean parkCar(Car newCar)
```

Begin your response at the top of a new page in the separate Free Response booklet and fill in the appropriate circle at the top of each page to indicate the question number. If there are multiple parts to this question, write the part letter with your response.

STOP

END OF EXAM

The Princeton Review®

Completely darken bubbles with a No. 2 pencil. If you make a mistake, be sure to erase mark completely. Erase all stray marks.

1.

YOUR NAME: (Print) _____ Last _____ First _____ M.I.

SIGNATURE: _____ DATE: ___/___/___

HOME ADDRESS: (Print) _____ Number and Street

_____ City _____ State _____ Zip Code

PHONE NO.: _____

IMPORTANT: Please fill in these boxes exactly as shown on the back cover of your test book.

2. TEST FORM

3. TEST CODE

4. REGISTRATION NUMBER

0	A	J	0	0	0	0	0	0	0	0	0
1	B	K	1	1	1	1	1	1	1	1	1
2	C	L	2	2	2	2	2	2	2	2	2
3	D	M	3	3	3	3	3	3	3	3	3
4	E	N	4	4	4	4	4	4	4	4	4
5	F	O	5	5	5	5	5	5	5	5	5
6	G	P	6	6	6	6	6	6	6	6	6
7	H	Q	7	7	7	7	7	7	7	7	7
8	I	R	8	8	8	8	8	8	8	8	8
9			9	9	9	9	9	9	9	9	9

5. YOUR NAME

First 4 letters of last name					FIRST INIT	MID INIT
A	A	A	A		A	A
B	B	B	B		B	B
C	C	C	C		C	C
D	D	D	D		D	D
E	E	E	E		E	E
F	F	F	F		F	F
G	G	G	G		G	G
H	H	H	H		H	H
I	I	I	I		I	I
J	J	J	J		J	J
K	K	K	K		K	K
L	L	L	L		L	L
M	M	M	M		M	M
N	N	N	N		N	N
O	O	O	O		O	O
P	P	P	P		P	P
Q	Q	Q	Q		Q	Q
R	R	R	R		R	R
S	S	S	S		S	S
T	T	T	T		T	T
U	U	U	U		U	U
V	V	V	V		V	V
W	W	W	W		W	W
X	X	X	X		X	X
Y	Y	Y	Y		Y	Y
Z	Z	Z	Z		Z	Z

6. DATE OF BIRTH

Month	Day		Year	
JAN				
FEB	0	0	0	0
MAR	1	1	1	1
APR	2	2	2	2
MAY	3	3	3	3
JUN		4	4	4
JUL		5	5	5
AUG		6	6	6
SEP		7	7	7
OCT		8	8	8
NOV		9	9	9
DEC				

The Princeton Review®

1. A B C D E	11. A B C D E	21. A B C D E	31. A B C D E
2. A B C D E	12. A B C D E	22. A B C D E	32. A B C D E
3. A B C D E	13. A B C D E	23. A B C D E	33. A B C D E
4. A B C D E	14. A B C D E	24. A B C D E	34. A B C D E
5. A B C D E	15. A B C D E	25. A B C D E	35. A B C D E
6. A B C D E	16. A B C D E	26. A B C D E	36. A B C D E
7. A B C D E	17. A B C D E	27. A B C D E	37. A B C D E
8. A B C D E	18. A B C D E	28. A B C D E	38. A B C D E
9. A B C D E	19. A B C D E	29. A B C D E	39. A B C D E
10. A B C D E	20. A B C D E	30. A B C D E	40. A B C D E