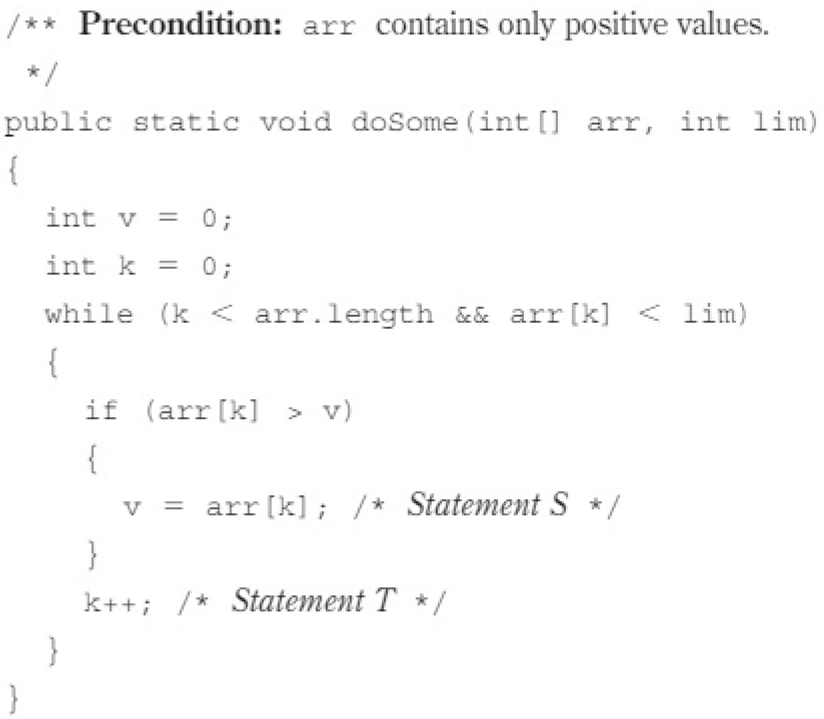
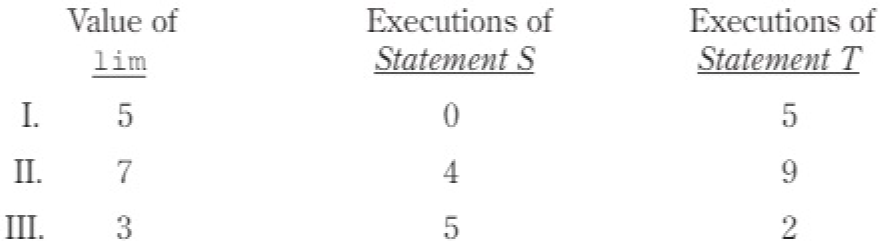
Round 1 - APCS A

1. Consider the following method.



```
/** Precondition: arr contains only positive values.
 */
public static void doSome(int[] arr, int lim)
{
  int v = 0;
  int k = 0;
  while (k < arr.length && arr[k] < lim)
  {
    if (arr[k] > v)
    {
      v = arr[k]; /* Statement S */
    }
    k++; /* Statement T */
  }
}
```

Assume that doSome is called and executes without error. Which of the following are possible combinations for the value of lim, the number of times *Statement* S is executed, and the number of times *Statement* T is executed?



	Value of lim	Executions of *Statement S*	Executions of *Statement T*
I.	5	0	5
II.	7	4	9
III.	3	5	2

(A) I only

(B) II only

(C) III only

(D) I and III only

(E) II and III only

Round 1 - APCS A

2. Consider the following code segment.

```
int[][] anArray = new int[10][8];

for (int j = 0; j < 8; j++)
{
     for (int k = 0; k < 10; k++)
     {
         anArray[j][k] = 5;
     }
}
```

The code segment causes an `ArrayIndexOutOfBoundsException` to be thrown. How many elements in `anArray` will be set to `5` before the exception is thrown?

(A) 0

(B) 8

(C) 9

(D) 64

(E) 80

3. Assume that `mat` has been declared as a 4×4 array of integers and has been initialized to contain all `1`s. Consider the following code segment.

```
int n = mat.length;
for (int j = 1; j < n; j++)
{
     for (int k = 1; k < n; k++)
     {
         mat[j][k] = mat[j - 1][k] + mat[j][k - 1];
     }
}
```

What is the value of `mat[2][2]` after the code segment has completed execution?

(A) `2`

(B) `3`

(C) `4`

(D) `6`

(E) `10`

4. The `Car` class will contain two string attributes for a car's make and model. The class will also contain a constructor.

```
public class Car
{
     /* missing code */
}
```

Which of the following replacements for `/*` *missing code* `*/` is the most appropriate implementation of the class?

(A)

```
public String make;
public String model;
public Car(String myMake, String myModel)
{ /* implementation not shown */ }
```

(B)

```
public String make;
public String model;
private Car(String myMake, String myModel)
{ /* implementation not shown */ }
```

(C)

```
private String make;
private String model;
public Car(String myMake, String myModel)
{ /* implementation not shown */ }
```

(D)

```
public String make;
private String model;
private Car(String myMake, String myModel)
( /* implementation not shown */ }
```

(E)

```
private String make;
private String model;
private Car(String myMake, String myModel)
{ /* implementation not shown */ }
```

5. The `Date` class below will contain three `int` attributes for day, month, and year, a constructor, and a `setDate` method. The `setDate` method is intended to be accessed outside the class.

```
public class Date
{
     /* missing code */
}
```

Which of the following replacements for `/*` *missing code* `*/` is the most appropriate implementation of the class?

Round 1 - APCS A

(A)
```
private int day;
private int month;
private int year;
private Date()
{ /* implementation not shown */ }
private void setDate(int d, int m, int y)
{ /* implementation not shown */ }
```

(B)
```
private int day;
private int month;
private int year;
public Date()
{ /* implementation not shown */ }
private void setDate(int d, int m, int y)
{ /* implementation not shown */ }
```

(C)
```
private int day;
private int month;
private int year;
public Date()
{ /* implementation not shown */ }
public void setDate(int d, int m, int y)
{ /* implementation not shown */ }
```

(D)
```
public int day;
public int month;
public int year;
private Date()
{ /* implementation not shown */ }
private void setDate(int d, int m, int y)
{ /* implementation not shown */ }
```

(E)
```
public int day;
public int month;
public int year;
public Date()
{ /* implementation not shown */ }
public void setDate(int d, int m, int y)
{ /* implementation not shown */ }
```

6. The `Player` class below will contain two `int` attributes and a constructor. The class will also contain a method `getScore` that can be accessed from outside the class.

```
public class Player
{
     /* missing code */
}
```

Which of the following replacements for `/*` *missing code* `*/` is the most appropriate implementation of the class?

(A)

```
private int score;
private int id;
private Player(int playerScore, int playerID)
{ /* implementation not shown */ }
private int getScore()
{ /* implementation not shown */ }
```

(B)

```
private int score;
private int id;
public Player(int playerScore, int playerID)
{ /* implementation not shown */ }
private int getScore()
{ /* implementation not shown */ }
```

(C)

```
private int score;
private int id;
public Player(int playerScore, int playerID)
{ /* implementation not shown */ }
public int getScore()
{ /* implementation not shown */ }
```

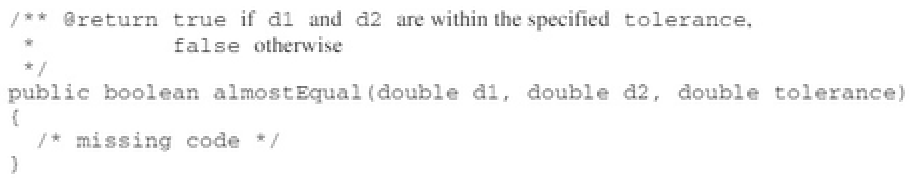
(D)

```
public int score;
public int id;
public Player(int playerScore, int playerID)
{ /* implementation not shown */ }
private int getScore()
{ /* implementation not shown */ }
```

(E)

```
public int score;
public int id;
public Player(int playerScore, int playerID)
{ /* implementation not shown */ }
public int getScore()
{ /* implementation not shown */ }
```

7. Consider the following method that is intended to determine if the double values d1 and d2 are close enough to be considered equal. For example, given a tolerance of 0.001, the values 54.32271 and 54.32294 would be considered equal.



```
/** @return true if d1 and d2 are within the specified tolerance,
 *          false otherwise
 */
public boolean almostEqual(double d1, double d2, double tolerance)
{
  /* missing code */
}
```

Which of the following should replace / * missing code * / so that almostEqual will work as intended?

(A) return (d1 - d2) <= tolerance;

(B) return ((d1 + d2) / 2) <= tolerance;

(C) return (d1 - d2) >= tolerance;

(D) return ((d1 + d2) / 2) >= tolerance;

(E) return Math.abs(d1 - d2) <= tolerance;

8. Consider the following class declarations.

```
public class Alpha
{
     private int answer()
     {
         return 10;
     }
}

public class Beta
{
     public double sample()
     {
         Alpha item = new Alpha();
         double temp = item.answer();
         return temp * 2.0;
     }
}
```

Which of the following best describes why an error occurs when the classes are compiled?

(A) The class `Alpha` does not have a defined constructor.

(B) The class `Alpha` must be declared as a subclass of `Beta`.

(C) The class `Beta` must be declared as a subclass of `Alpha`.

(D) The `answer` method cannot be accessed from a class other than `Alpha`.

(E) The result of the method call `item.answer()` cannot be assigned to a variable of type `double`.

9. A bear is an animal and a zoo contains many animals, including bears. Three classes Animal, Bear, and Zoo are declared to represent animal, bear, and zoo objects. Which of the following is the most appropriate set of declarations?

```
public class Animal extends Bear
{
  ...
}
```

(A)

```
public class Zoo
{
  private Animal[] myAnimals;
  ...
}
public class Bear extends Animal
{
  ...
}
```

(B)

```
public class Zoo
{
  private Animal[] myAnimals;
  ...
}
public class Animal extends Zoo
{
```

(C)

```
  private Bear myBear;
  ...
}
public class Bear extends Animal, Zoo
```

(D)

```
{
  ...
}
public class Bear extends Animal implements Zoo
```

(E)

```
{
  ...
}
```

10. Consider the following code segment.

```
ArrayList<String> colors = new ArrayList<String>();

colors.add("Red");
colors.add("Orange");
colors.set(1, "Yellow");
colors.add(1, "Green");
colors.set(colors.size() - 1, "Blue");
colors.remove(0);
System.out.println(colors);
```

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

(A) `[Red, Orange]`

(B) `[Red, Green]`

(C) `[Yellow, Blue]`

(D) `[Green, Blue]`

(E) `[Blue, Yellow]`

11. Consider the following statement, which is intended to create an `ArrayList` named `theater_club` to store elements of type `Student`. Assume that the `Student` class has been properly defined and includes a no-parameter constructor.

```
ArrayList<Student> theater_club = new /* missing code */;
```

Which choice can replace `/*` *missing code* `*/` so that the statement compiles without error?

(A) `Student()`

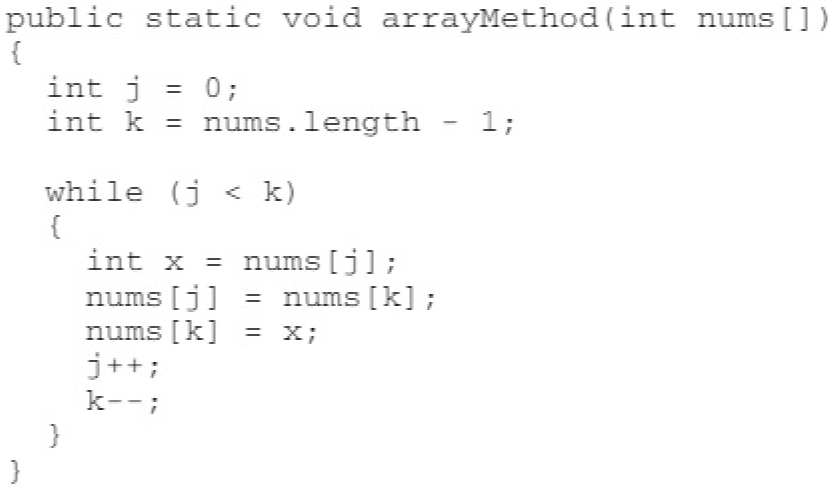
(B) `Student ArrayList()`

(C) `ArrayList(Student)`

(D) `ArrayList<Student>()`

(E) `ArrayList<theater_club>()`

12. Consider the following method.



```
public static void arrayMethod(int nums[])
{
  int j = 0;
  int k = nums.length - 1;

  while (j < k)
  {
    int x = nums[j];
    nums[j] = nums[k];
    nums[k] = x;
    j++;
    k--;
  }
}
```

Which of the following describes what the method arrayMethod() does to the array nums?

(A) The array nums is unchanged.

(B) The first value in nums is copied to every location in the array.

(C) The last value in nums is copied to every location in the array.

(D) The method generates an ArrayIndexOutOfBoundsException.

(E) The contents of the array nums are reversed.

Round 1 - APCS A

13. Consider the following method, between, which is intended to return true if x is between lower and upper, inclusive, and false otherwise.

```
// precondition:  lower <= upper

// postcondition: returns true if x is between lower and upper,

//              inclusive; otherwise, returns false

public boolean between(int x, int lower, int upper)

{

  /* missing code */

}
```

Which of the following can be used to replace /* missing code */ so that between will work as intended?

(A) return (x <= lower) && (x >= upper);

(B) return (x <= lower) || (x >= upper);

(C) return lower <= x <= upper;

(D) return (x >= lower) && (x <= upper);

(E) return (x >= lower) || (x <= upper);

14. A two-dimensional array `arr` is to be created with the following contents.

```
boolean[][] arr = {{false, true, false},
                   {false, false, true}};
```

Which of the following code segments can be used to correctly create and initialize `arr` ?

(A)
```
boolean arr[][] = new boolean[2][3];
arr[0][1] = true;
arr[1][2] = true;
```

(B)
```
boolean arr[][] = new boolean[2][3];
arr[1][2] = true;
arr[2][3] = true;
```

(C)
```
boolean arr[][] = new boolean[3][2];
arr[0][1] = true;
arr[1][2] = true;
```

(D)
```
boolean arr[][] = new boolean[3][2];
arr[1][0] = true;
arr[2][1] = true;
```

(E)
```
boolean arr[][] = new boolean[3][2];
arr[2][1] = true;
arr[3][2] = true;
```

15. **The following question is based on the following incomplete declaration of the class BoundedIntArray and its constructor definitions.**

A BoundedintArray represents an indexed list of integers. In a BoundedIntArray the user can specify a size, in which case the indices range from 0 to size - 1. The user can also specify the lowest index, low, in which case the indices can range from low to low + size - 1.

```
public class BoundedIntArray
{
  private int[] myItems;   // storage for the list
  private int myLowIndex;  // lowest index


  public BoundedIntArray(int size)
  {
    myItems = new int[size];
    myLowIndex = 0;
  }


  public BoundedIntArray(int size, int low)
  {
    myItems = new int[size];
    myLowIndex = low;
  }
  // other methods not shown
}
```

Consider the following statements.

```
BoundedIntArray arrl = new BoundedIntArray(100, 5);
```

Round 1 - APCS A

```
BoundedIntArray arr2 = new BoundedIntArray(100);
```

Which of the following best describes arrl and arr2 after these statements?

(A) arrl and arr2 both represent lists of integers indexed from 0 to 99.

(B) arrl and arr2 both represent lists of integers indexed from 5 to 104.

(C) arrl represents a list of integers indexed from 0 to 104, and arr2 represents a list of integers indexed from 0 to 99.

(D) arrl represents a list of integers indexed from 5 to 99, and arr2 represents a list of integers indexed from 0 to 99.

(E) arrl represents a list of integers indexed from 5 to 104, and arr2 represents a list of integers indexed from 0 to 99.

16. **The following question is based on the following incomplete declaration of the class BoundedIntArray and its constructor definitions.**

A BoundedintArray represents an indexed list of integers. In a BoundedIntArray the user can specify a size, in which case the indices range from 0 to size - 1. The user can also specify the lowest index, low, in which case the indices can range from low to low + size - 1.

```
public class BoundedIntArray
{
  private int[] myItems;   // storage for the list
  private int myLowIndex;  // lowest index

  public BoundedIntArray(int size)
  {
    myItems = new int[size];
    myLowIndex = 0;
  }

  public BoundedIntArray(int size, int low)
  {
    myItems = new int[size];
    myLowIndex = low;
  }
  // other methods not shown
}
```

Which of the following is the best reason for declaring the data fields myItems and myLowIndex to be private rather than public?

Round 1 - APCS A

(A) This permits BoundedIntArray objects to be initialized and modified.

(B) This permits BoundedIntArray methods to be written and tested before code that uses a BoundedIntArray is written.

(C) This helps to prevent clients of the BoundedIntArray class from writing code that would need to be modified if the implementation of BoundedIntArray were changed.

(D) This prevents compile-time errors whenever public methods are called that access the private data fields.

(E) This prevents run-time errors whenever public methods are called that access the private data fields.

17. Consider the following three class declarations.

```
public class ClassOne
{
     public void methodA()
     { /* implementation not shown */ }

     public void methodB()
     { /* implementation not shown */ }
}

public class ClassTwo
{
     public void methodA()
     { /* implementation not shown */ }
}

public class ClassThree extends ClassOne
{
     public void methodB()
     { /* implementation not shown */ }
}
```

The following declarations occur in a method in another class.

```
ClassOne one = new ClassOne();
ClassTwo two = new ClassTwo();
ClassThree three = new ClassThree();
/* missing method call */
```

Which of the following replacements for `/*` *missing method call* `*/` will cause a compile-time error?

(A) `one.methodA();`

(B) `two.methodA();`

(C) `two.methodB();`

(D) `three.methodA();`

(E) `three.methodB();`

Round 1 - APCS A

18. A car dealership needs a program to store information about the cars for sale. For each car, they want to keep track of the following information: number of doors (2 or 4), whether the car has air conditioning, and its average number of miles per gallon. Which of the following is the best object-oriented program design?

(A) Use one class, Car, with three instance variables:
int numDoors, boolean hasAir, and
double milesPerGallon.

(B) Use four unrelated classes: Car, Doors, AirConditioning, and MilesPerGallon.

(C) Use a class Car with three subclasses: Doors, AirConditioning, and MilesPerGallon.

(D) Use a class Car, with a subclass Doors, with a subclass AirConditioning, with a subclass MilesPerGallon.

(E) Use three classes: Doors, AirConditioning, and MilesPerGallon, each with a subclass Car.

Round 1 - APCS A

19. Consider the following two methods that appear within a single class.

```
public void changeIt(int[] list, int num)
{
  list = new int[5];
  num = 0;

  for (int x = 0; x < list.length; x++)
    list[x] = 0;
}

public void start()
{
  int[] nums = {1, 2, 3, 4, 5};
  int value = 6;

  changeIt(nums, value);

  for (int k = 0; k < nums.length; k++)
    System.out.print(nums[k] + " ");

  System.out.print(value);
}
```

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

Round 1 - APCS A

(A) 0 0 0 0 0 0

(B) 0 0 0 0 0 6

(C) 1 2 3 4 5 6

(D) 1 2 3 4 5 0

(E) changeIt will throw an exception.

20. Consider the following class definitions.

```
public class ClassA
{
     public String getValue()
     {
         return "A";
     }
     public void showValue()
     {
         System.out.print(getValue());
     }
}
public class ClassB extends ClassA
{
     public String getValue()
     {
         return "B";
     }
}
```

The following code segment appears in a class other than `ClassA` or `ClassB`.

```
ClassA obj = new ClassB();
obj.showValue();
```

What, if anything, is printed when the code segment is executed?

(A) `A`

(B) `B`

(C) `AB`

(D) `BA`

(E) Nothing is printed because the code does not compile.

21. Which of the following code segments produces the output `"987654321"` ?

Round 1 - APCS A

(A)

```
int num = 10;
while (num > 0)
{
     System.out.print(num);
     num--;
}
```

(B)

```
int num = 10;
while (num >= 0)
{
     System.out.print(num);
     num--;
}
```

(C)

```
int num = 10;
while (num > 1)
{
     num--;
     System.out.print(num);
}
```

(D)

```
int num = 10;
while (num >= 1)
{
     num--;
     System.out.print(num);
}
```

(E)

```
int num = 0;
while (num <= 9)
{
     System.out.print(10 - num);
     num++;
}
```

Round 1 - APCS A

22. Consider the following methods.

```
/** Precondition: a > 0 and b > 0 */
public static int methodOne(int a, int b)
{
     int loopCount = 0;
     for (int i = 0; i < a / b; i++)
     {
         loopCount++;
     }
     return loopCount;
}
/** Precondition: a > 0 and b > 0 */
public static int methodTwo(int a, int b)
{
     int loopCount = 0;
     int i = 0;
     while (i < a)
     {
         loopCount++;
         i += b;
     }
     return loopCount;
}
```

Which of the following best describes the conditions under which `methodOne` and `methodTwo` return the same value?

(A) When `a` and `b` are both even

(B) When `a` and `b` are both odd

(C) When `a` is even and `b` is odd

(D) When `a % b` is equal to zero

(E) When `a % b` is equal to one

Round 1 - APCS A

23. Consider the following incomplete method, which is intended to return `true` if the value of `y` is between the values of the other two parameters and `false` otherwise.

```
/** Precondition: x, y, and z have 3 different values. */
public static boolean compareThree(int x, int y, int z)
{
     return /* missing condition */ ;
}
```

The following table shows the results of several calls to `compareThree`.

Call	**Result**
`compareThree(4, 5, 6)`	`true`
`compareThree(6, 5, 4)`	`true`
`compareThree(5, 4, 6)`	`false`
`compareThree(3, 4, 4)`	violates precondition

Which of the following can be used to replace `/*` *missing condition* `*/` so that `compareThree` will work as intended when called with parameters that satisfy its precondition?

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

(B) `(x > y) && (y > z)`

(C) `(x > y) || (y > z)`

(D) `(x > y) == (y > z)`

(E) `(x > y) != (y > z)`

24. Which of the following is a reason to use an ArrayList instead of an array?

(A) An ArrayList allows faster access to its *k*th item than an array does.

(B) An ArrayList always uses less memory than an array does.

(C) An ArrayList can store objects and an array can only store primitive types.

(D) An ArrayList resizes itself as necessary when items are added, but an array does not.

(E) An ArrayList provides access to the number of items it stores, but an array does not.

25. Consider the following declarations.

```
int valueOne, valueTwo;
```

Assume that valueOne and valueTwo have been initialized. Which of the following evaluates to true if valueOne and valueTwo contain the same value?

(A) valueOne.equals((Object) valueTwo)

(B) valueOne == valueTwo

(C) valueOne.compareTo((Object) valueTwo) == 0

(D) valueOne.compareTo(valueTwo) == 0

(E) valueOne.equals(valueTwo)

26. Consider the following code segment.

```
int x = /* some integer value */ ;

int y = /* some integer value */ ;



boolean result = (x < y);



result = ( (x >= y) && !result );
```

Which of the following best describes the conditions under which the value of result will be true after the code segment is executed?

(A) Only when x < y

(B) Only when x >= y

(C) Only when x and y are equal

(D) The value will always be true.

(E) The value will never be true.

Round 1 - APCS A

27. Consider the following code segment. The code is intended to read nonnegative numbers and compute their product until a negative number is read; however, it does not work as intended. (Assume that the readInt method correctly reads the next number from the input stream.)

```
int k = 0;

int prod = 1;



while (k >= 0)

{

  System.out.print("enter a number: ");

  k = readInt(); // readInt reads the next number from input

  prod = prod * k;

}



System.out.println("product: " + prod);
```

Which of the following best describes the error in the program?

(A) The variable prod is incorrectly initialized.

(B) The while condition always evaluates to false.

(C) The while condition always evaluates to true.

(D) The negative number entered to signal no more input is included in the product.

(E) If the user enters a zero, the computation of the product will be terminated prematurely.

28. Consider the following code segment.

```
int num = 1;
int count = 0;
while (num <= 10)
{
     if (num % 2 == 0 && num % 3 == 0)
     {
         count++;
     }
     num++;
}
```

What value is stored in the variable `count` as a result of executing the code segment?

Round 1 - APCS A

(A) `1`

(B) `3`

(C) `5`

(D) `7`

(E) `8`

29. Consider the following method.

```
public void changeIt(int[] arr, int index, int newValue)
{
     arr[index] += newValue;
}
```

Which of the following code segments, if located in a method in the same class as `changeIt`, will cause the array `myArray` to contain `{0, 5, 0, 0}` ?

(A)
```
int[] myArray = new int[4];
changeIt(myArray, 1, 5);
```

(B)
```
int[] myArray = new int[4];
changeIt(myArray, 2, 5);
```

(C)
```
int[] myArray = new int[4];
changeIt(myArray, 5, 1);
```

(D)
```
int[] myArray = new int[5];
changeIt(myArray, 1, 4);
```

(E)
```
int[] myArray = new int[5];
changeIt(myArray, 1, 5);
```

30. Consider the following code segment, which is intended to declare and initialize the two-dimensional (2D) `String` array `things`.

```
/* missing code */ = {{"spices", "garlic", "onion", "pepper"},
                     {"clothing", "hat", "scarf", "gloves"},
                     {"plants", "tree", "bush", "flower"},
                     {"vehicles", "car", "boat", "airplane"}};
```

Which of the following could replace `/*` *missing code* `*/` so that `things` is properly declared?

(A) `new String[][] things`

(B) `new(String[][]) things`

(C) `String[] String[] things`

(D) `String[][] things`

(E) `[][]String things`

Round 1 - APCS A

31. Consider the following class declarations. Assume that each class has a no-argument constructor.

```
public class Food
{ /* implementation not shown */ }

public class Snack extends Food
{ /* implementation not shown */ }

public class Pizza extends Snack
{ /* implementation not shown */ }
```

Which of the following declarations will compile without error?

(A) `Food tacos = new Snack();`

(B) `Pizza cheesePizza = new Snack();`

(C) `Pizza sausagePizza = new Food();`

(D) `Snack pretzel = new Food();`

(E) `String Snack = new Pizza();`

32. When designing a class hierarchy, which of the following should be true of a superclass?

(A) A superclass should contain the data and functionality that are common to all subclasses that inherit from the superclass.

(B) A superclass should be the largest, most complex class from which all other subclasses are derived.

(C) A superclass should contain the data and functionality that are only required for the most complex class.

(D) A superclass should have public data in order to provide access for the entire class hierarchy.

(E) A superclass should contain the most specific details of the class hierarchy.

33. Consider the following code segment.

```
int count = 5;
while (count < 100)
{
     count = count * 2;
}
count = count + 1;
```

What will be the value of `count` as a result of executing the code segment?

(A) `100`

(B) `101`

(C) `160`

(D) `161`

(E) `321`

Round 1 - APCS A

34. Consider the following class definition.

```
public class Example
{
     private int x;
     // Constructor not shown.
}
```

Which of the following is a correct header for a method of the `Example` class that would return the value of the `private` instance variable `x` so that it can be used in a class other than `Example` ?

(A) `private int getX()`

(B) `private void getX()`

(C) `public int getX()`

(D) `public void getX()`

(E) `public void getX(int x)`

Round 1 - APCS A

35. Consider the following two static methods, where f2 is intended to be the iterative version of f1.

```
public static int f1(int n)
{
 if (n < 0)
 {
   return 0;
 }
 else
 {
   return (f1(n - 1) + n * 10);
 }
}
public static int f2(int n)
{
  int answer = 0;
  while (n > 0)
  {
    answer = answer + n * 10;
    n--;
  }

  return answer;
}
```

The method f2 will always produce the same results as f1 under which of the following conditions?

Round 1 - APCS A

I. n < 0

II. n = 0

III. n > 0

(A) I only

(B) II only

(C) III only

(D) II and III only

(E) I, II, and III

36. Consider the following code segment.

```
int value = 15;

while (value < 28)

{

  System.out.println(value);

  value++;

}
```

What are the first and last numbers output by the code segment?

Round 1 - APCS A

(A)

First	Last
15	27

(B)

First	Last
15	28

(C)

First	Last
16	27

(D)

First	Last
16	28

(E)

First	Last
16	29

Round 1 - APCS A

37. Consider the following method.

```
public int getTheResult(int n)

{

int product = 1;

for (int number = 1; number < n; number++)

{

if (number % 2 == 0)

product *= number;

}

return product;

}
```

What value is returned as a result of the call getTheResult(8) ?

(A) 48

(B) 105

(C) 384

(D) 5040

(E) 40320

Round 1 - APCS A

The following questions refer to the code from the GridWorld case study. A copy of the code is provided below.

Appendix B — Testable API

info.gridworld.grid.Location class (implements Comparable)

public Location(int r, int c)

constructs a location with given row and column coordinates

public int getRow()

returns the row of this location

public int getCol()

returns the column of this location

public Location getAdjacentLocation(int direction)

returns the adjacent location in the direction that is closest to direction

public int getDirectionToward(Location target)

returns the closest compass direction from this location toward target

public boolean equals(Object other)

returns true if other is a Location with the same row and column as this location; false otherwise public int hashCode()

returns a hash code for this location

public int compareTo(Object other)

returns a negative integer if this location is less than other, zero if the two locations are equal, or a positive integer if this location is greater than other. Locations are ordered in row-major order.

Precondition: other is a Location object.

public String toString()

returns a string with the row and column of this location, in the format (row, col)

Compass directions:

public static final int NORTH = 0;

public static final int EAST = 90;

public static final int SOUTH = 180;

public static final int WEST = 270;

public static final int NORTHEAST = 45;

public static final int SOUTHEAST = 135;

public static final int SOUTHWEST = 225;

public static final int NORTHWEST = 315;

Turn angles:

public static final int LEFT = -90;

public static final int RIGHT = 90;

Round 1 - APCS A

public static final int HALF_LEFT = -45;

public static final int HALF_RIGHT = 45;

public static final int FULL_CIRCLE = 360;

public static final int HALF_CIRCLE = 180;

public static final int AHEAD = 0;

info.gridworld.grid.Grid<E> interface

int getNumRows()

returns the number of rows, or -1 if this grid is unbounded

int getNumCols()

returns the number of columns, or -1 if this grid is unbounded

boolean isValid(Location loc)

returns true if loc is valid in this grid, false otherwise

Precondition: loc is not null

E put(Location loc, E obj)

puts obj at location loc in this grid and returns the object previously at that location (or null if the location was previously unoccupied).

Precondition: (1) loc is valid in this grid (2) obj is not null

E remove(Location loc)

Round 1 - APCS A

removes the object at location loc from this grid and returns the object that was removed (or null if the location is unoccupied)

Precondition: loc is valid in this grid

E get(Location loc)

returns the object at location loc (or null if the location is unoccupied)

Precondition: loc is valid in this grid

ArrayList<Location> getOccupiedLocations()

returns an array list of all occupied locations in this grid

ArrayList<Location> getValidAdjacentLocations(Location loc)

returns an array list of the valid locations adjacent to loc in this grid

Precondition: loc is valid in this grid

ArrayList<Location> getEmptyAdjacentLocations(Location loc)

returns an array list of the valid empty locations adjacent to loc in this grid

Precondition: loc is valid in this grid

ArrayList<Location> getOccupiedAdjacentLocations(Location loc) returns an array list of the valid occupied locations adjacent to loc in this grid

Precondition: loc is valid in this grid

ArrayList<E> getNeighbors(Location loc)

returns an array list of the objects in the occupied locations adjacent to loc in this grid

Precondition: loc is valid in this grid

info.gridworld.actor.Actor class

public Actor()

constructs a blue actor that is facing north

public Color getColor()

returns the color of this actor

public void setColor(Color newColor)

sets the color of this actor to newColor

public int getDirection()

returns the direction of this actor, an angle between 0 and 359 degrees

public void setDirection(int newDirection)

Round 1 - APCS A

sets the direction of this actor to the angle between 0 and 359 degrees that is equivalent to newDirection

public Grid<Actor> getGrid()

returns the grid of this actor, or null if this actor is not contained in a grid

public Location getLocation()

returns the location of this actor, or null if this actor is not contained in a grid

public void putSelfInGrid(Grid<Actor> gr, Location loc)

puts this actor into location loc of grid gr. If there is another actor at loc, it is removed.

Precondition: (1) This actor is not contained in a grid (2) loc is valid in gr

public void removeSelfFromGrid()

removes this actor from its grid.

Precondition: this actor is contained in a grid

public void moveTo(Location newLocation)

moves this actor to newLocation. If there is another actor at newLocation, it is removed.

Precondition: (1) This actor is contained in a grid (2) newLocation is valid in the grid of this actor

public void act()

reverses the direction of this actor. Override this method in subclasses of Actor to define types of actors with different behavior

public String toString()

returns a string with the location, direction, and color of this actor

info.gridworld.actor.Rock class (extends Actor)

public Rock()

constructs a black rock

public Rock(Color rockColor)

constructs a rock with color rockColor

public void act()

overrides the act method in the Actor class to do nothing

info.gridworld.actor.Flower class (extends Actor)

public Flower()

constructs a pink flower

Round 1 - APCS A

public Flower(Color initialColor)

constructs a flower with color initialColor

public void act()

causes the color of this flower to darken

38. A CornerBug behaves like a Bug except that a CornerBug makes all turns at right angles rather than 45-degree angles. Of the following, which is the best design for the CornerBug class?

(A) Create an abstract class called RightAngleBug that is a Bug that only turns 90 degrees, and then create a class CornerBug that inherits from RightAngleBug.

(B) Create an interface called RightTurn that includes the specification of a turnRight method, and then create a class CornerBug that implements RightTurn.

(C) Create a class CornerBug that inherits from Bug and adds a constructor that has an int parameter that determines if the bug should turn 90 degrees or 45 degrees.

(D) Create a class CornerBug that inherits from Bug and overrides the Bug turn method to turn the bug 90 degrees instead of 45 degrees.

(E) Create an interface CornerBug that includes the definition of a turnRight method that is automatically used by the Bug act method for objects that are instantiated as CornerBug objects.

39. Assume that an array of integer values has been declared as follows and has been initialized.

```
int[] arr = new int[10];
```

Which of the following code segments correctly interchanges the value of `arr[0]` and `arr[5]` ?

(A)
```
arr[0] = 5;
arr[5] = 0;
```

(B)
```
arr[0] = arr[5];
arr[5] = arr[0];
```

(C)
```
int k = arr[5];
arr[0] = arr[5];
arr[5] = k;
```

(D)
```
int k = arr[0];
arr[0] = arr[5];
arr[5] = k;
```

(E)
```
int k = arr[5];
arr[5] = arr[0];
arr[0] = arr[5];
```

Round 1 - APCS A

40. The following method is intended to print the number of digits in the parameter `num`.

```
public int numDigits(int num)
{
     int count = 0;
     while (/* missing condition */)
     {
         count++;
         num = num / 10;
     }
     return count;
}
```

Which of the following can be used to replace `/*` *missing condition* `*/` so that the method will work as intended?

(A) `count != 0`

(B) `count > 0`

(C) `num >= 0`

(D) `num != 0`

(E) `num == 0`

Round 1 - APCS A

41. Consider the following method.

```
public int mystery(int num)

{

int x = num;

while (x > 0)

{

if (x / 10 % 2 == 0)

return x;

x = x / 10;

}

return x;

}
```

What value is returned as a result of the call mystery(1034) ?

(A) 4

(B) 10

(C) 34

(D) 103

(E) 1034

42. Consider the following code segment.

```
int k = 0;

while (k < 10)

{

  System.out.print((k % 3) + " ");

  if ((k % 3) == 0)

    k = k + 2;

  else

    k++;

}
```

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

(A) 0 2 1 0 2

(B) 0 2 0 2 0 2

(C) 0 2 1 0 2 1 0

(D) 0 2 0 2 0 2 0

(E) 0 1 2 1 2 1 2

43. Consider the following code segment, which is intended to print the sum of all the odd integers from 0 up to and including 101.

```
int r = 0;
int sum = 0;
/* missing loop header */
{
     if (r % 2 == 1)
     {
         sum += r;
     }
     r++;
}
System.out.println(sum);
```

Which of the following could replace `/* missing loop header */` to ensure that the code segment will work as intended?

(A) `while (r <= 100)`

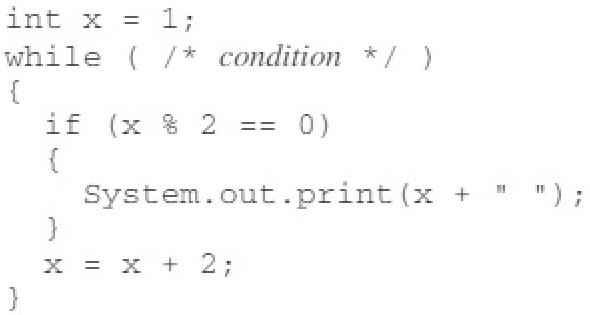
(B) `while (sum <= 100)`

(C) `while (r < 101)`

(D) `while (r <= 101)`

(E) `while (sum <= 101)`

44. Consider the following code segment.



```
int x = 1;
while ( /* condition */ )
{
  if (x % 2 == 0)
  {
    System.out.print(x + " ");
  }
  x = x + 2;
}
```

The following conditions have been proposed to replace /* *condition* */ in the code segment.

I. x < 0

II. x <= 1

III. x < 10

For which of the conditions will nothing be printed?

(A) I only

(B) II only

(C) I and II only

(D) I and III only

(E) I, II, and III

45. Consider the following code segment.

```
int a = 24;

int b = 30;

while (b != 0)

{

  int r = a % b;

  a = b;

  b = r;

}


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

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

(A) 0

(B) 6

(C) 12

(D) 24

(E) 30

46. Consider the following code segment. Assume that `num3 > num2 > 0`.

```
int num1 = 0;
int num2 = /* initial value not shown */;
int num3 = /* initial value not shown */;
while (num2 < num3)
{
     num1 += num2;
     num2++;
}
```

Which of the following best describes the contents of `num1` as a result of executing the code segment?

(A) The product of `num2` and `num3`

(B) The product of `num2` and `num3 - 1`

(C) The sum of `num2` and `num3`

(D) The sum of all integers from `num2` to `num3`, inclusive

(E) The sum of all integers from `num2` to `num3 - 1`, inclusive

47. The following question refer to the following information.

Consider the following data field and method. Method maxHelper is intended to return the largest value among the first numVals values in an array; however, maxHelper does not work as intended.

```
     private int[] nums;


     // precondition: 0 < numVals <= nums.length

     private int maxHelper(int numVals)

     {

Line 1:   int max = maxHelper(numVals - 1);


Line 2:   if (max > nums[numVals - 1])

            return max;

          else

            return nums[numVals - 1];

      }
```

Which of the following best describes the conditions under which maxHelper does not work as intended?

(A) When numVals is 1

(B) When numVals is even

(C) When the elements of nums are in nonincreasing order

(D) When the elements of nums are in nondecreasing order

(E) Method maxHelper never works as intended.

Round 1 - APCS A

48. The following question refer to the following information.

Consider the following data field and method. Method maxHelper is intended to return the largest value among the first numVals values in an array; however, maxHelper does not work as intended.

```
      private int[] nums;


      // precondition: 0 < numVals <= nums.length

      private int maxHelper(int numVals)

      {

Line 1:   int max = maxHelper(numVals - 1);


Line 2:   if (max > nums[numVals - 1])

            return max;

          else

            return nums[numVals - 1];

       }
```

Which of the following corrects the method maxHelper so that it works as intended?

(A) Insert the following statement before Line 1.

```
if (numVals == 0)

  return numVals;
```

(B) Insert the following statement before Line 1.

```
if (numVals == 1
  return nums[0];
```

(C) Insert the following statement between Line 1 and Line 2.

```
if (numVals == 0)
  return numVals;
```

(D) Insert the following statement between Line 1 and Line 2.

```
if (numVals == 1)
  return nums[0];
```

(E) Insert the following statement between Line 1 and Line 2.

```
if (numVals < 2)
  return numVals;
```

Round 1 - APCS A

49. Consider the following statement, which is intended to create an `ArrayList` named `values` that can be used to store `Integer` elements.

```
/* missing code */ = new ArrayList<>();
```

Which of the following can be used to replace `/*` *missing code* `*/` so that the statement compiles without error?

I. `ArrayList values`
II. `ArrayList<int> values`
III. `ArrayList<Integer> values`

(A) I only

(B) II only

(C) III only

(D) I and III only

(E) II and III only

50. Consider the following statement, which is intended to create an `ArrayList` named `years` that can be used to store elements both of type `Integer` and of type `String`.

```
/* missing code */ = new ArrayList();
```

Which of the following can be used to replace `/*` *missing code* `*/` so that the statement compiles without error?

(A) `ArrayList years`

(B) `ArrayList years()`

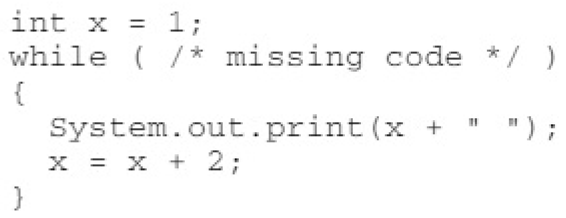
(C) `ArrayList years[]`

(D) `ArrayList<Integer> years`

(E) `ArrayList<String> years`

Round 1 - APCS A

51. Consider the following code segment.



```
int x = 1;
while ( /* missing code */ )
{
  System.out.print(x + " ");
  x = x + 2;
}
```

Consider the following possible replacements for /* missing code */.

I. x < 6

II. x != 6

III. x < 7

Which of the proposed replacements for /* missing code */ will cause the code segment to print only the values 1 3 5?

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

52. Consider the following code segment.

```
int k = 0;
/* missing loop header */
{
     k++;
     System.out.print(k + " ");
}
```

Which of the following can be used as a replacement for `/*` *missing loop header* `*/` so that the code segment prints out the string `"1 2 3 4 "`?

(A) `while (k < 3)`
(B) `while (k < 4)`
(C) `while (k < 5)`
(D) `while (k <= 4)`
(E) `while (k <= 5)`

Round 1 - APCS A

53. Consider the following code segment.

```
int val = 48;
int div = 6;
while ((val % 2 == 0) && div > 0)
{
     if (val % div == 0)
     {
         System.out.print(val + " ");
     }
     val /= 2;
     div--;
}
```

What is printed when the code segment is executed?

(A) `48 12 6`

(B) `48 12 6 3`

(C) `48 12 6 3 1`

(D) `48 24 12 6`

(E) `48 24 12 6 3`

Round 1 - APCS A

54. Consider the following method.

```
public static int mystery(int[] arr)
{
     int count = 0;
     int curr = arr[arr.length - 1];

     for (int value : arr)
     {
         if (value > curr)
         {
            count = count + 1;
         }
         else
         {
            count = count - 1;
         }
         curr = value;
     }

     return count;
}
```

The following code segment appears in another method of the same class.

```
int[] arr = {4, 14, 15, 3, 14, 18, 19};
System.out.println(mystery(arr));
```

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

(A) `-7`

(B) `-6`

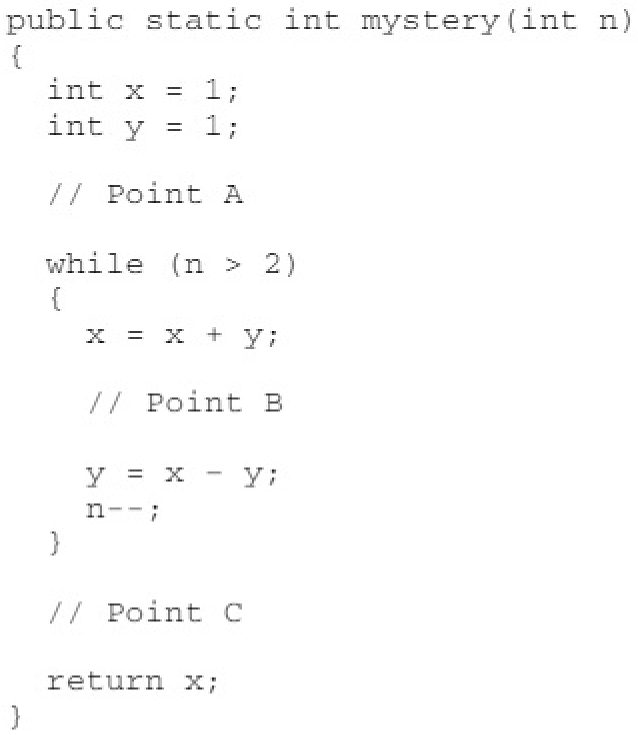
(C) `3`

(D) `5`

(E) `7`

Round 1 - APCS A

Directions: Select the choice that best fits each statement. The following question(s) refer to the following method



```
public static int mystery(int n)
{
  int x = 1;
  int y = 1;

  // Point A

  while (n > 2)
  {
    x = x + y;

    // Point B

    y = x - y;
    n--;
  }

  // Point C

  return x;
}
```

55. What value is returned as a result of the call mystery (6)?

(A) 1

(B) 5

(C) 6

(D) 8

(E) 13

56. Which of the following is true of method mystery ?

(A) x will sometimes be 1 at // Point B.

(B) x will never be 1 at // Point C.

(C) n will never be greater than 2 at // Point A.

(D) n will sometimes be greater than 2 at // Point C.

(E) n will always be greater than 2 at // Point B.

Round 1 - APCS A

57. Consider the following data field and method.

```
private int[][] mat;

public int mystery(int r, int c)

{

  if (r != 0 || c != 0)

  {

    return (mat[r][c] + mystery(r - 1, c - 1));

  }

  else

  {

    return mat[r][c];

  }

}
```

Assume that mat is the 2-D array shown below.

	0	**1**	**2**	**3**
0	0	1	2	3
1	4	5	6	7
2	8	9	10	11
3	12	13	14	15

What value is returned as a result of the call mystery(2, 3)?

(A) 1

(B) 11

(C) 17

(D) 18

(E) No value is returned because mystery throws an ArrayIndexOutOfBoundsException.

Round 1 - APCS A

58. Consider the following method.

```
   // precondition: arr contains no duplicates;
   //             the elements in arr are in sorted order;
   //             0 ≤ low ≤ arr.length; low - 1 ≤ high < arr.length
   public static int mystery(int[] arr, int low, int high, int num)
   {
     int mid = (low + high) / 2;

     if (low > high)
     {
       return low;
     }
     else if (arr[mid] < num)
     {
       return mystery(arr, mid + 1, high, num);
     }
     else if (arr[mid] > num)
     {
       return mystery(arr, low, mid - 1, num);
     }
     else // arr{mid] == num
     {
       return mid;
}
}
```

How many calls to mystery (including the initial call) are made as a result of the call mystery(arr, 0, arr.length - 1, 14) if arr is the following array?

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
arr	11	13	25	26	29	30	31	32

(A) 1
(B) 2
(C) 4
(D) 7
(E) 8

59. A student has created a `Car` class. The class contains variables to represent the following.

A `String` variable called `color` to represent the color of the car
An `int` variable called `year` to represent the year the car was made
A `String` variable called `make` to represent the manufacturer of the car
A `String` variable called `model` to represent the model of the car

The object `vehicle` will be declared as type `Car`.

Which of the following descriptions is accurate?

(A) An instance of the `vehicle` class is `Car`.
(B) An instance of the `Car` object is `vehicle`.
(C) An attribute of the `year` object is `int`.
(D) An attribute of the `vehicle` object is `color`.
(E) An attribute of the `Car` instance is `vehicle`.

60. Consider the following method.

```
public void numberCheck(int maxNum)
{
  int typeA = 0;
  int typeB = 0;
  int typeC = 0;

  for (int k = 1; k <= maxNum; k++)
  {
    if (k % 2 == 0 && k % 5 == 0)
     typeA++;
    if (k % 2 == 0)
      typeB++;
    if (k % 5 == 0)
      typeC++;
  }

  System.out.println(typeA + " " + typeB + " " + typeC);
}
```

What is printed as a result of the call `numberCheck(50)` ?

(A) 5 20 5

(B) 5 20 10

(C) 5 25 5

(D) 5 25 10

(E) 30 25 10

Round 1 - APCS A

61. Consider the following code segment.

```
int[] numbers = new int[5];
numbers[0] = 2;
numbers[1] = numbers[0] + 1;
numbers[numbers[0]] = numbers[1];

for (int x = 3; x < numbers.length; x++)
{
     numbers[x] = numbers[x - 1] * 2;
}
```

Which of the following represents the contents of the array `numbers` after the code segment is executed?

(A) `{2, 3, 0, 0, 0}`

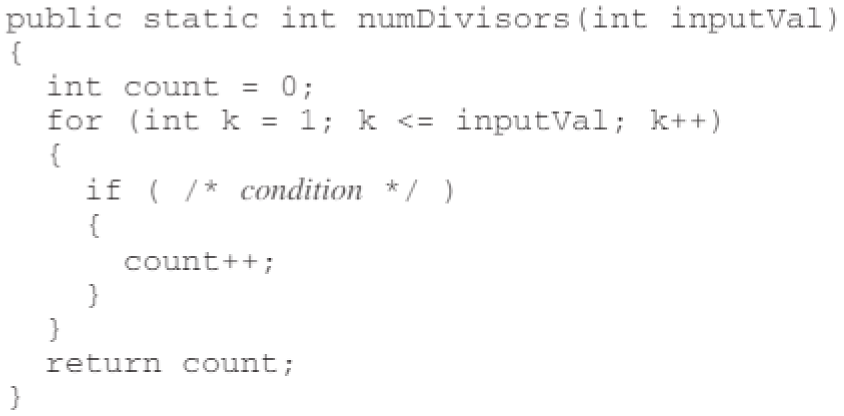
(B) `{2, 3, 1, 2, 4}`

(C) `{2, 3, 3, 6, 9}`

(D) `{2, 3, 3, 6, 12}`

(E) `{2, 4, 8, 16, 32}`

62. Consider the following incomplete method, which is intended to return the number of integers that evenly divide the integer inputVal. Assume that inputVal is greater than 0.



```
public static int numDivisors(int inputVal)
{
  int count = 0;
  for (int k = 1; k <= inputVal; k++)
  {
    if ( /* condition */ )
    {
      count++;
    }
  }
  return count;
}
```

Which of the following can be used to replace / * *condition* * / so that numDivisors will work as intended?

(A) inputVal % k == 0

(B) k % inputVal == 0

(C) inputVal % k != 0

(D) inputVal / k == 0

(E) k / inputVal > 0

Round 1 - APCS A

63. Consider the following declaration of the class NumSequence, which has a constructor that is intended to initialize the instance variable seq to an ArrayList of numberOfValues random floating-point values in the range [0.0, 1.0).

```
public class NumSequence

{

  private ArrayList<Double> seq;


  // precondition: numberOfValues > 0

  // postcondition: seq has been initialized to an ArrayList of

  //              length numberOfValues; each element of seq

  //              contains a random Double in the range [0.0, 1.0)

  public NumSequence(int numberOfValues)

  {

    /* missing code */

  }

}
```

Which of the following code segments could be used to replace /* *missing code* */ so that the constructor will work as intended?

I.
```
      ArrayList<Double> seq = new ArrayList<Double>();

  for (int k = 0; k < numberOfValues; k++)

    seq.add(new Double(Math.random()));
```

II.
```
     seq = new ArrayList<Double>();

  for (int k = 0; k < numberOfValues; k++)
```

seq.add(new Double(Math.random()));

III. ArrayList<Double> temp = new ArrayList<Double>();

for (int k = 0; k < numberOfValues; k++)

temp.add(new Double(Math.random()));

seq = temp;

(A) II only

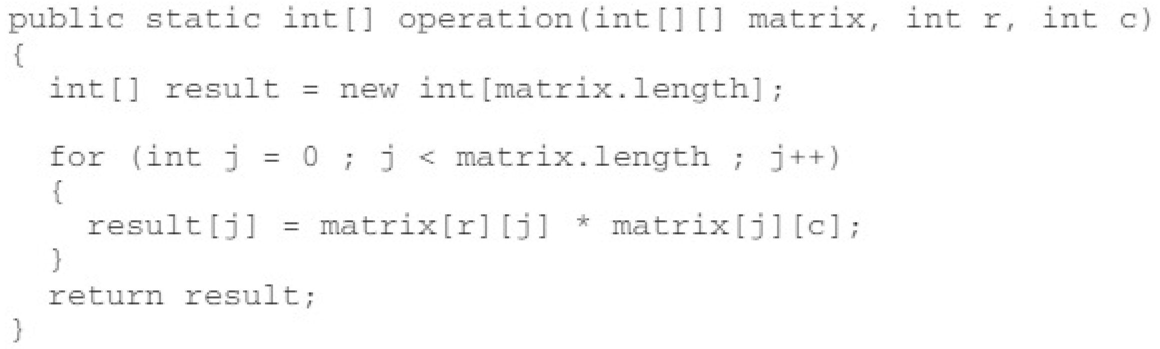
(B) III only

(C) I and II

(D) I and III

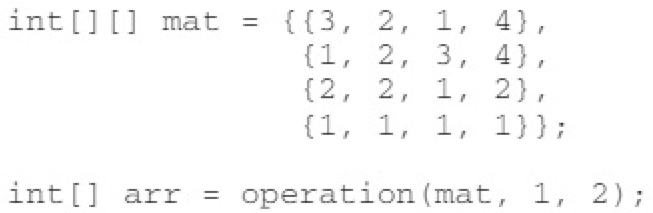
(E) II and III

64. Consider the following method.



```
public static int[] operation(int[][] matrix, int r, int c)
{
  int[] result = new int[matrix.length];

  for (int j = 0 ; j < matrix.length ; j++)
  {
    result[j] = matrix[r][j] * matrix[j][c];
  }
  return result;
}
```

The following code segment appears in another method in the same class.



```
int[][] mat = {{3, 2, 1, 4},
               {1, 2, 3, 4},
               {2, 2, 1, 2},
               {1, 1, 1, 1}};

int[] arr = operation(mat, 1, 2);
```

Which of the following represents the contents of arr as a result of executing the code segment?

(A) {6, 4, 2, 4}
(B) {1, 6, 3, 4}
(C) {4, 3, 6, 1}
(D) {4, 4, 2, 2}
(E) {2, 2, 4, 4}

65. A student has created an `OrderedPair` class to represent points on an xy-plane. The class contains the following.

An `int` variable called `x` to represent an x-coordinate.
An `int` variable called `y` to represent a y-coordinate.
A method called `printXY` that will print the values of `x` and `y`.

The object `origin` will be declared as type `OrderedPair`.

Which of the following descriptions is accurate?

(A) `origin` is an instance of the `printXY` method.
(B) `origin` is an instance of the `OrderedPair` class.
(C) `origin` is an instance of two `int` objects.
(D) `OrderedPair` is an instance of the `origin` object.
(E) `printXY` is an instance of the `OrderedPair` class.

66. Consider the following method.

```
public int[] addNum(int[] array, int first, int second, int num)
{
     int[] newArray = new int[array.length];
     newArray[first] = array[first] + num;
     newArray[second] = array[second] + num;
     return newArray;
}
```

Which of the following code segments, appearing in the same class as the `addNum` method, will result in `array2` having the contents `{0, 0, 13, 0, 9, 0, 0}` ?

(A)
```
int[] array1 = {5, 2, 8, 6, 4, 3, 9};
int[] array2 = addNum(array1, 2, 4, 5);
```

(B)
```
int[] array1 = {-5, -5, 13, 0, 9, 0, 0};
int[] array2 = addNum(array1, 2, 4, 5);
```

(C)
```
int[] array1 = {5, 2, 8, 6, 4, 3, 9};
int[] array2 = addNum(array1, 3, 5, 5);
```

(D)
```
int[] array1 = {5, 8, 2, 4, 6, 3, 9};
int[] array2 = addNum(array1, 2, 4, 5);
```

(E)
```
int[] array1 = {0, -5, 8, 0, 9, 0, 0};
int[] array2 = addNum(array1, 2, 4, 5);
```

Round 1 - APCS A

67. Consider the following method.

```
public static int getValue(int[] data, int j, int k)
{
     return data[j] + data[k];
}
```

Which of the following code segments, when appearing in another method in the same class as `getValue`, will print the value `70` ?

(A)
```
int arr = {40, 30, 20, 10, 0};
System.out.println(getValue(arr, 1, 2));
```

(B)
```
int[] arr = {40, 30, 20, 10, 0};
System.out.println(getValue(arr, 1, 2));
```

(C)
```
int[] arr = {50, 40, 30, 20, 10};
System.out.println(getValue(arr, 1, 2));
```

(D)
```
int arr = {40, 30, 20, 10, 0};
System.out.println(getValue(arr, 2, 1));
```

(E)
```
int arr = {50, 40, 30, 20, 10};
System.out.println(getValue(arr, 2, 1));
```

68. Consider the following code segment.

```
int a = 10;
int b = 5 * 2;
System.out.print(a == b);
```

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

(A) `5`

(B) `10`

(C) `10 == 10`

(D) `true`

(E) `false`

69. Consider the following code segment.

```
int num = 1;
while (num < 5)
{
     System.out.print("A");
     num += 2;
}
```

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

(A) `A`

(B) `AA`

(C) `AAA`

(D) `AAAA`

(E) `AAAAA`

70. Assume that a two-dimensional (2D) array `arr` of `String` objects with 3 rows and 4 columns has been properly declared and initialized.

Which of the following can be used to print the elements in the four corner elements of `arr` ?

(A) `System.out.print(arr[0, 0] + arr[0, 3] + arr[2, 0] + arr[2, 3]);`

(B) `System.out.print(arr[1, 1] + arr[1, 4] + arr[3, 1] + arr[3, 4]);`

(C) `System.out.print(arr[0][0] + arr[0][2] + arr[3][0] + arr[3][2]);`

(D) `System.out.print(arr[0][0] + arr[0][3] + arr[2][0] + arr[2][3]);`

(E) `System.out.print(arr[1][1] + arr[1][4] + arr[3][1] + arr[3][4]);`

71. Consider the following code segment, where `letters` is a two-dimensional (2D) array that contains possible letters. The code segment is intended to print `"DIG"`.

```
String[][] letters = {{"A", "B", "C"},
                      {"D", "E", "F"},
                      {"G", "H", "I"}};
System.out.println( /* missing code */ );
```

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

(A) `letters[2][1] + letters[3][3] + letters[3][1]`

(B) `letters[2][0] + letters[2][2] + letters[1][0]`

(C) `letters[1][2] + letters[3][3] + letters[1][3]`

(D) `letters[1][0] + letters[2][2] + letters[2][0]`

(E) `letters[0][1] + letters[2][2] + letters[0][2]`

Round 1 - APCS A

72. Which of the following code segments will print all multiples of 5 that are greater than 0 and less than 100 ?

```
I. for (int k = 1; k < 100; k++)
{
   if (k % 5 == 0)
   {
      System.out.print(k + " ");
   }
}

II. for (int k = 1; k < 100; k++)
{
   if (k / 5 == 0)
   {
      System.out.print(k + " ");
   }
}

III. int k = 5;
while (k < 100)
{
   System.out.print(k + " ");
   k = k + 5;
}
```

(A) I only

(B) II only

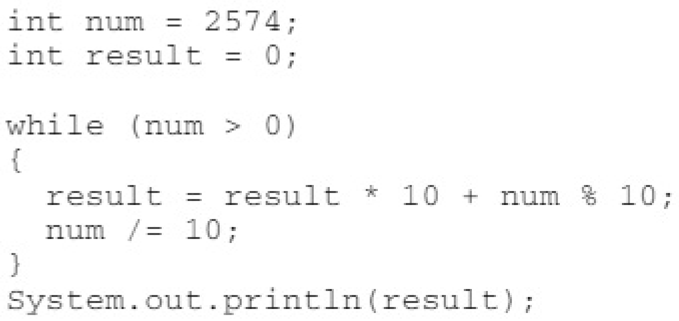
(C) III only

(D) I and III

(E) II and III

Round 1 - APCS A

73. Consider the following code segment.



```
int num = 2574;
int result = 0;

while (num > 0)
{
  result = result * 10 + num % 10;
  num /= 10;
}
System.out.println(result);
```

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

(A) 2

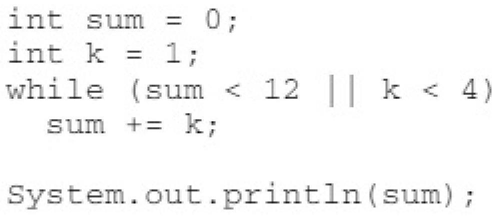
(B) 4

(C) 18

(D) 2574

(E) 4752

74. Consider the following code segment.



```
int sum = 0;
int k = 1;
while (sum < 12 || k < 4)
  sum += k;

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

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

(A) 6

(B) 10

(C) 12

(D) 15

(E) Nothing is printed due to an infinite loop.

Round 1 - APCS A

75. Consider the following code segment, where `nums` is a two-dimensional (2D) array of integers. The code segment is intended to print `"test1234"`.

```
System.out.print("test" + nums[0][0] + nums[1][0] + nums[1][1] +
nums[0][1]);
```

Which of the following code segments properly declares and initializes `nums` so that the code segment works as intended?

(A) `int[][] nums = {{1, 2}, {3, 4}};`

(B) `int[][] nums = {{1, 2}, {4, 3}};`

(C) `int[][] nums = {{1, 3}, {2, 4}};`

(D) `int[][] nums = {{1, 4}, {2, 3}};`

(E) `int[][] nums = {{1, 4}, {3, 2}};`

76. Consider the following code segment.

```
double a = 1.1;

double b = 1.2;



if ((a + b) * (a - b) != (a * a) - (b * b))

{

  System.out.println("Mathematical error!");

}
```

Which of the following best describes why the phrase "Mathematical error!" would be printed?

(Remember that mathematically $(a + b) * (a - b) = a^2 - b^2$.)

(A) Precedence rules make the if condition true.

(B) Associativity rules make the if condition true.

(C) Roundoff error makes the if condition true.

(D) Overflow makes the if condition true.

(E) A compiler bug or hardware error has occurred.

Round 1 - APCS A

77. Consider the following recursive method.

```
/** Precondition: 0 <= numVals <= nums.length */
public static int mystery(int[] nums, int v, int numVals)
{
     if (numVals == 0)
     {
         return 0;
     }
     else if (v == nums[numVals - 1])
     {
         return 1 + mystery(nums, v, numVals - 1);
     }
     else
     {
         return mystery(nums, v, numVals - 1);
     }
}
```

Which of the following best describes the value returned by the call `mystery(nums, v, nums.length)` ?

(A) The value `0` is returned.

(B) The value `1` is returned.

(C) The number of times that `v` occurs in `nums` is returned.

(D) The number of times that `numVals` occurs in `nums` is returned.

(E) Nothing is returned. A runtime error occurs because of infinite recursion.

Round 1 - APCS A

78. Consider the following class declarations.

```
public class Shoe
{
     private String shoeBrand;
     private String shoeModel;

     public Shoe(String brand, String model)
     {
         shoeBrand = brand;
         shoeModel = model;
     }

     // No other constructors
}

public class Boot extends Shoe
{
     private double heelHeight;

     public Boot(String brand, String model, double height)
     {
         /* missing implementation */
     }
}
```

Which of the following should be used to replace `/*` *missing implementation* `*/` so that all instance variables are initialized with parameters?

(A)
```
shoeBrand = brand;
shoeModel = model;
heelHeight = height;
```

(B)
```
super();
heelHeight = height;
```

(C)
```
super(brand, model);
```

(D)
```
heelHeight = height;
super(brand, model);
```

(E)
```
super(brand, model);
heelHeight = height;
```

79. Consider the following method.

```
public static String abMethod(String a, String b)
{
     int x = a.indexOf(b);
     while (x >= 0)
     {
         a = a.substring(0, x) + a.substring(x + b.length());
         x = a.indexOf(b);
     }
     return a;
}
```

What, if anything, is returned by the method call `abMethod("sing the song", "ng")` ?

(A) `"si"`

(B) `"si the so"`

(C) `"si the song"`

(D) `"sig the sog"`

(E) Nothing is returned because a `StringIndexOutOfBoundsException` is thrown.

80. Consider the following method.

```
public int someCode(int a, int b, int c)

{

  if ((a < b) && (b < c))

    return a;

  if ((a >= b) && (b >= c))

    return b;

  if ((a == b) || (a == c) || (b == c))

    return c;

}
```

Which of the following best describes why this method does not compile?

(A) The reserved word return cannot be used in the body of an if statement.

(B) It is possible to reach the end of the method without returning a value.

(C) The if statements must have else parts when they contain `return` statements.

(D) Methods cannot have multiple `return` statements.

(E) The third if statement is not reachable.

81. A student has created a `Song` class. The class contains the following variables.

A `String` variable called `artist` to represent the artist name
A `String` variable called `title` to represent the song title
A `String` variable called `album` to represent the album title

The object `happyBirthday` will be declared as type `Song`.

Which of the following statements is true?

(A) `artist`, `title`, and `album` are instances of the `Song` class.

(B) `happyBirthday` is an instance of three `String` objects.

(C) `happyBirthday` is an instance of the `Song` class.

(D) `Song` is an instance of the `happyBirthday` object.

(E) `Song` is an instance of three `String` objects.

Round 1 - APCS A

82. Consider the following method.

```
public static void sort(String[] arr)

{

  for (int pass = arr.length - 1; pass >= 1; pass--)

  {

    String large = arr[0];

    int index = 0;

    for (int k = 0; k <= pass; k++)

    {

      if ((arr[k].compareTo(large)) > 0)

      {

        large = arr[k];

        index = k;

      }

    }

    arr[index] = arr[pass];

    arr[pass] = large;

  }

}
```

Assume arr is the following array.

"Ann"	"Mike"	"Walt"	"Lisa"	"Shari"	"Jose"	"Mary"	“Bill”

What is the intermediate value of arr after two iterations of the outer for loop in the call sort(arr)?

(A)

"Ann"	"Mike"	"Walt"	"Lisa"	"Shari"	"Jose"	"Mary"	“Bill”

(B)

"Ann"	"Mike"	"Lisa"	"Shari"	"Jose"	"Mary"	“Bill”	"Walt"

(C)

"Ann"	“Bill”	"Jose"	"Lisa"	"Mary"	"Mike"	"Shari"	"Walt"

(D)

"Ann"	"Mike"	“Bill”	"Lisa"	"Mary"	"Jose"	"Shari"	"Walt"

(E)

"Walt"	"Shari"	"Ann"	"Lisa"	"Mike"	"Jose"	"Mary"	“Bill”

83. Consider the following code segment.

```
int[][] values = {{1, 2, 3}, {4, 5, 6}};
int x = 0;
for (int j = 0; j < values.length; j++)
{
    for (int k = 0; k < values[0].length; k++)
    {
        if (k == 0)
        {
           values[j][k] *= 2;
        }
        x += values[j][k];
    }
}
```

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

(A) 7

(B) 17

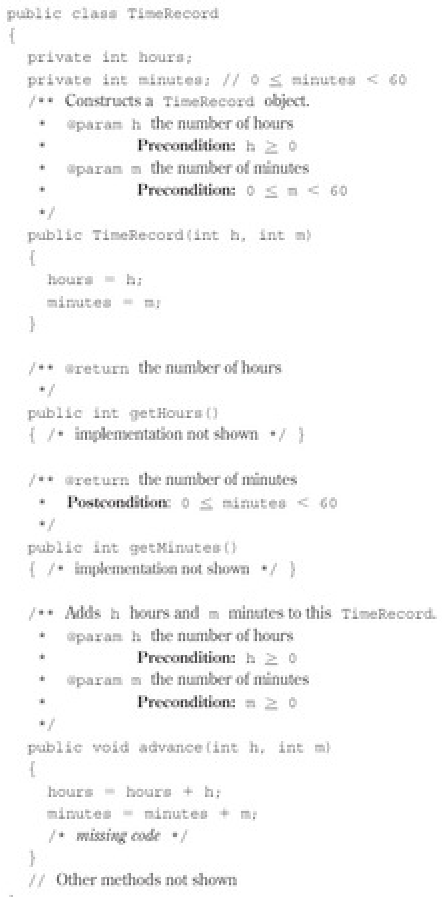
(C) 21

(D) 26

(E) 27

Round 1 - APCS A

Directions: Select the choice that best fits each statement. The following question(s) refer to the following incomplete class declaration.



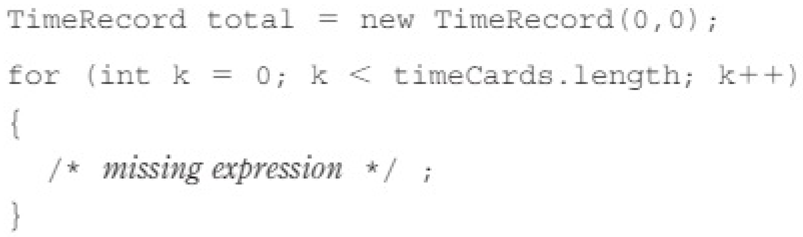
```
public class TimeRecord
{
  private int hours;
  private int minutes; // 0 ≤ minutes < 60
  /** Constructs a TimeRecord object.
   *  @param h the number of hours
   *           Precondition: h ≥ 0
   *  @param m the number of minutes
   *           Precondition: 0 ≤ m < 60
   */
  public TimeRecord(int h, int m)
  {
    hours = h;
    minutes = m;
  }

  /** @return the number of hours
   */
  public int getHours()
  { /* implementation not shown */ }

  /** @return the number of minutes
   *  Postcondition: 0 ≤ minutes < 60
   */
  public int getMinutes()
  { /* implementation not shown */ }

  /** Adds h hours and m minutes to this TimeRecord.
   *  @param h the number of hours
   *           Precondition: h ≥ 0
   *  @param m the number of minutes
   *           Precondition: m ≥ 0
   */
  public void advance(int h, int m)
  {
    hours = hours + h;
    minutes = minutes + m;
    /* missing code */
  }
  // Other methods not shown
```

Round 1 - APCS A

84. Consider the following declaration that appears in a class other than TimeRecord.

TimeRecord [ ] timeCards = new TimeRecord [100] ;

Assume that timeCards has been initialized with TimeRecord objects. Consider the following code segment that is intended to compute the total of all the times stored in timeCards.



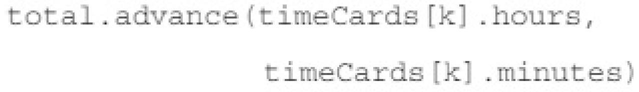
```
TimeRecord total = new TimeRecord(0,0);
for (int k = 0; k < timeCards.length; k++)
{
   /* missing expression */ ;
}
```

Which of the following can be used to replace / * *missing expression* * / so that the code segment will work as intended?

(A) timeCards [k] .advance ( )

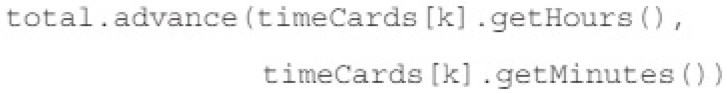
(B) total += timeCards [k] .advance ( )

(C)



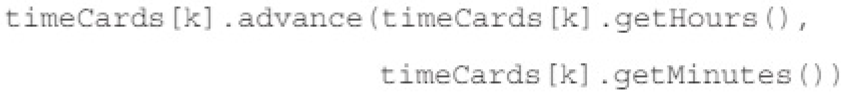
```
total.advance(timeCards[k].hours,
              timeCards[k].minutes)
```

(D)



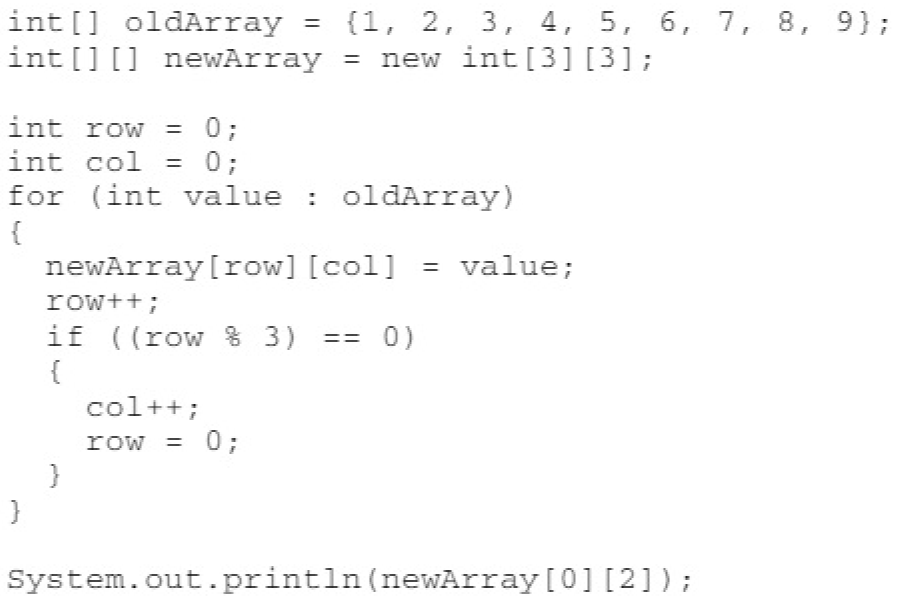
```
total.advance(timeCards[k].getHours(),
              timeCards[k].getMinutes())
```

(E)



```
timeCards[k].advance(timeCards[k].getHours(),
                     timeCards[k].getMinutes())
```

85. Consider the following code segment.



```
int[] oldArray = {1, 2, 3, 4, 5, 6, 7, 8, 9};
int[][] newArray = new int[3][3];

int row = 0;
int col = 0;
for (int value : oldArray)
{
  newArray[row][col] = value;
  row++;
  if ((row % 3) == 0)
  {
    col++;
    row = 0;
  }
}

System.out.println(newArray[0][2]);
```

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

(A) 3

(B) 4

(C) 5

(D) 7

(E) 8

86. Consider the following code segment.

```
int[] oldArray = {1, 2, 3, 4, 5, 6, 7, 8, 9};

int[][] newArray = new int[3][3];

int row = 0; int col = 0;

for (int index = 0; index < oldArray.length; index++)

{

newArray[row][col] = oldArray[index]; row++;

if ((row % 3) == 0)

{

col++;

row = 0;

}

}

System.out.println(newArray[0][2]);
```

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

(A) 3

(B) 4

(C) 5

(D) 7

(E) 8

87. Consider the following statement.

```
boolean x = (5 < 8) == (5 == 8);
```

What is the value of `x` after the statement has been executed?

Round 1 - APCS A

(A) 3

(B) 5

(C) 8

(D) true

(E) false

88. Consider the following Boolean expression in which the `int` variables `x` and `y` have been properly declared and initialized.

```
(x <= 10) == (y > 25)
```

Which of the following values for `x` and `y` will result in the expression evaluating to `true` ?

(A) `x = 8` and `y = 25`

(B) `x = 10` and `y = 10`

(C) `x = 10` and `y = 30`

(D) `x = 15` and `y = 30`

(E) `x = 25` and `y = 30`

89. Consider the following method.

```
public boolean checkIndexes(double[][] data, int row, int col)
{
     int numRows = data.length;
     if (row < numRows)
     {
         int numCols = data[0].length;
         return col < numCols;
     }
     else
     {
         return false;
     }
}
```

Consider the following variable declaration and initialization, which appears in a method in the same class as `checkIndexes`.

```
double[][] table = new double[5][6];
```

Which of the following method calls returns a value of `true` ?

(A) `checkIndexes(table, 4, 5)`

(B) `checkIndexes(table, 4, 6)`

(C) `checkIndexes(table, 5, 4)`

(D) `checkIndexes(table, 5, 6)`

(E) `checkIndexes(table, 6, 5)`

Round 1 - APCS A

90. Consider the following code segment.

```
int num1 = 0;
int num2 = 3;
while ((num2 != 0) && ((num1 / num2) >= 0))
{
  num1 = num1 + 2;
  num2 = num2 - 1;
}
```

What are the values of numl and num2 after the while loop completes its execution?

(A) num1 = 0, num2 = 3

(B) num1 = 8, num2 = -1

(C) num1 = 4, num2 = 1

(D) num1 = 6, num2 = 0

(E) The loop will never complete its execution because a division by zero will generate an ArithmeticException.

91. Consider the following code segment.

```
int k = 1;
while (k < 20)
{
  if ((k % 3) == 1)
    System.out.print(k + " ");

  k++;
}
```

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

Round 1 - APCS A

(A) 2 5 8 11 14 17

(B) 3 6 9 12 15 18

(C) 1 4 7 10 13 16 19

(D) 1 3 5 7 9 11 13 15 17 19

(E) 2 4 6 8 10 12 14 16 18 20

92. When designing classes, which of the following would be the best reason to use inheritance?

(A) Inheritance allows you to write applications that require fewer base and super classes.

(B) Inheritance allows the creation of a subclass that can use the methods of its superclass without rewriting the code for those methods.

(C) Inheritance allows for data encapsulation, while noninherited classes do not allow for data encapsulation.

(D) Inheritance reduces the number of polymorphic structures encapsulated in applications.

(E) Inheritance guarantees that the applications will compile and execute much more quickly.