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. Which of the following will print a number less than 5?

   I. `System.out.println (24 / 5 % 3 * 2);`
   II. `System.out.println (12 / 3 * 2 + 1);`
   III. `System.out.println ( 1 + 4 % 3 * 2);`

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

2. What output is generated by the following line of code:

```
System.out.println("Simon says, \n\\\"insert phrase here\"//");
```

   (A)
```
Simon says, \n \\ insert phrase here//
```
   (B)
```
Simon says, \n \\ "insert phrase here"//
```
   (C)
```
Simon says, \n "insert phrase here"
```
   (D)
```
Simon says,
  \"insert phrase here"/
```
   (E)
```
Simon says,
  \"insert phrase here"//
```

GO ON TO THE NEXT PAGE.

3. Consider the following class with the numbers added for reference:

```
public class Tree
{
  private String name;
  private double height;
  private int rateOfGrowth;

  Tree(String n, double h, int r)
  {
    name = n;
    height = h;
    rateOfGrowth = r;
  }
  Tree(double h, int r)
  {
    height = h;
    rateOfGrowth = r;
  }
  Tree(String n)
  {
    name = n;
  }
  public String toString()
  {
    return name + " can grow up to " + height + " feet high, at a rate of " + rateOfGrowth +
    " inches per year";
  }
}
```

Which of the following code excerpts in a client program would cause an error?

(A)
```
Tree elm = new Tree("Elm", 60, 36);
System.out.println(elm.toString());
```

(B)
```
Tree riverBirch = new Tree("River Birch", 40.0, (int)13.0);
System.out.println(riverBirch.toString());
```

(C)
```
Tree redMaple = new Tree(60.0, 18);
System.out.println(redMaple.toString());
```

(D)
```
Tree redwood = new Tree("Redwood", 300, 24.0);
System.out.println(redwood.toString());
```

(E)
```
Tree sequoia = new Tree("Sequoia");
System.out.println(sequoia.toString());
```

GO ON TO THE NEXT PAGE.

Questions 4–5 refer to the class SalesRep.

```
public class SalesRep
{
   private int idNum;
   private String Name;
   private int ytdSales;

   SalesRep(int i, String n, int ytd)
   {
      idNum = i;
      name = n;
      ytdSales = ytd;
   }

   public int getYtdSales() {return ytdSales;}
}
```

4. A client method, computeBonus, will return a salesRep bonus computed by multiplying his ytdSales by a percentage.

```
/** Precondition: SalesRep s has ytdSales >= 0
 * @param s a SalesRep
 * @param percentage represents what percent of the ytdSales represents the bonus
 * @return amount of bonus for the SalesRep (ytdSales * bonus)
 */

public static double computeBonus(SalesRep s, double percentage)
{    /* missing code */    }
```

Which replacement for /* missing code */ is correct?

(A) `return ytdSales() * percentage;`
(B) `return getYtdSales() * percentage;`
(C) `return s.ytdSales() * percentage;`
(D) `return s.getYtdSales() * percentage;`
(E) `return s.getYtdSales() * s.percentage;`

GO ON TO THE NEXT PAGE.

5. An `ArrayList` was created to store a `SalesRep` object for every salesperson in the XYZ company. Below is the declaration for that `ArrayList`.

```
ArrayList<SalesRep> list1 = new ArrayList<SalesRep>();
```

The company decided to pay each `SalesRep` a bonus since the company had a very profitable year. Management wishes to project the total of that payout, but to do so, must first calculate the total sales for the company by adding together the `ytdSales` from each `SalesRep`. Which code excerpt will compute the `ytdSales` for the company?

(A)
```
double sum = 0;
for (ArrayList r : list1)
{  sum += r.get.getYtdSales(); }
```

(B)
```
double sum = 0;
for (SalesRep r.get : list1)
{  sum += r.getYtdSales(); }
```

(C)
```
double sum = 0;
for (SalesRep r : list1)
{    sum += r.getYtdSales(); }
```

(D)
```
double sum = 0;
for (int i = 0; i < list1.size(); i++)
{    sum += SalesRep.get(i).getYtdSales();}
```

(E)
```
double sum = 0;
for (int i = 0; i <= list1.size(); i++)
{    sum += list1.get(i).getYtdSales();}
```

6. Which of the following will print after the following code is executed?

```
String str = new String("superstar");
System.out.print(str.substring (1, 3) + " ");
str.substring(1);
System.out.print(str.substring (1, 3) + " ");
str.substring(1);
System.out.print(str.substring (1, 3) + " ");
```

(A) `su up pe`
(B) `up pe er`
(C) `sup upe per`
(D) `up up up`
(E) A `StringIndexOutOfBoundsException` will occur

GO ON TO THE NEXT PAGE.

7. A chess game must take turns allowing black and white players to move on the board. The player is indicated by the following variable where `true` indicates black and `false` indicates white:

```
boolean isBlack;
```

At the end of each player's turn, the variable `isBlack` must alternate between `true` and `false` to indicate the next player's turn. Consider the following code examples, then determine those that would accomplish this purpose.

I.
```
isBlack = !isBlack;
```

II.
```
if (!isBlack)
    isBlack = true
else
    isBlack = false;
```

III.
```
if (isBlack)
    isBlack = false;
```

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

8. The following variables are declared, but their contents are unknown.

```
String str1 = new String("/*unknown value*/");
String str2 = new String("/*unknown value*/");
```

Consider the following decision statements and determine the statement that would fail to compare whether the value of `str1` is the same as the value of `str2`.

(A) `if (str1 == str2)`
(B) `if (str1.equals(str2))`
(C) `if (str1.compareTo(str2) == 0)`
(D) `if (str1.substring(0).equals(str2.substring(0,str2.length())))`
(E) `if (str1.length() == str2.length() && str1.indexOf(str2) == 0)`

9. Given the following boolean expression: `!(p || (q || !r))`

Which of the following conditions would result in an evaluation as `true`?

(A) `p = true  q = true  r = true`
(B) `p = true  q = false  r = true`
(C) `p = false  q = true  r = false`
(D) `p = false  q = false  r = false`
(E) `p = false  q = false  r = true`

GO ON TO THE NEXT PAGE.

10. Which boolean expression and values demonstrate a short-circuit evaluation?

(A) `p || !(q && r)` `p = true` `q = true` `r = true`
(B) `p || (q && !r)` `p = false` `q = true` `r = false`
(C) `p || !(q || r)` `p = false` `q = true` `r = true`
(D) `p && q && r` `p = true` `q = true` `r = true`
(E) `p && !(q && r)` `p = true` `q = true` `r = true`

11. Assume that `p` and `q` are boolean variables and have been properly initialized.

```
!(!p || q) || !(p || !q)
```

Which of the following best describe the result of evaluating the expression above?

(A) `true` always
(B) `false` always
(C) `true` only if `p` is `true`
(D) `true` only if `q` is `true`
(E) `true` only if `p` and `q` have opposite truth values

12. What output is generated by the following code excerpt:

```
int count = 0;
String star = "*";
for (int i = 1; i < 11; i++)
   for (int j = 10; j > 1; j -= 2)
   {
      star += "**";
      count++;
   }
System.out.print("\n" + count + " " + star.length());
```

(A) `50  100`
(B) `50  101`
(C) `51  100`
(D) `51  101`
(E) `90  181`

GO ON TO THE NEXT PAGE.

13. What output is generated by the following code excerpt:

```
for (int i = 1; i <= 5; i++)
{
   for (int j = 1; j < i; j++)
   {
      System.out.print("- ");
   }
   for (int j = i; j <= 5; j++)
   {
      System.out.print("* ");
   }
   System.out.println();
}
```

(A)
```
- - - - *
- - - * *
- - * * *
- * * * *
* * * * *
```

(B)
```
* * * * *
- * * * *
- - * * *
- - - * *
- - - - *
```

(C)
```
* - - - -
* * - - -
* * * - -
* * * * -
* * * * *
```

(D)
```
* * * * *
* * * * -
* * * - -
* * - - -
* - - - -
```

(E)
```
- - - - -
- - - - *
- - - * *
- - * * *
- * * * *
```

GO ON TO THE NEXT PAGE.

14. Consider the following code segments and determine those that would produce the same output.

I.
```
int sum = 0;
for (int i = 1; i < 3; i++)
{
      sum += 2 * i + 1;
}
System.out.println(sum);
```

II.
```
int sum = 0;
for (int i = 1; i <= 5; i++)
{
      if (i % 2 == 1)
            sum += i;
}
System.out.println(sum);
```

III.
```
int i = 5;
int sum = i;
while (i > 1)
{
      i -= 2;
      sum += i;
}
System.out.println(sum);
```

(A) I and II only
(B) I and III only
(C) II and III only
(D) I, II and III
(E) All three outputs are different.

GO ON TO THE NEXT PAGE.

15. Consider the following code segment:

```
/**
 * @param number is initialized with a positive integer value
 * @return the sum of odd integers between 1 and number
*/

public static int sumOdds(int number)
{
    int sum = 0;
    for (/* missing code */)
    {
        sum += k;
    }
    return sum;
}
```

Which of the following replacements for /* missing code */ will satisfy the conditions of the method?

I. `int k = 1; k <= number; k++`
II. `int k = 1; k <= number; k += 2`
III. `int k = number; k >= 1; k -= 2`

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

GO ON TO THE NEXT PAGE.

16. Using the following method to find the index of the largest value in an array. Choose the replacement(s) for /* some code */ that will accomplish the task as described.

```
/** Precondition: arr is initialized with integer values and is not empty
 * Integer.MIN_VALUE is a static constant containing the value –2147483648
 * @param arr the array to be processed
 * @return the location of the largest value in the array; if
 *         the largest value is stored in more than one element, return the
 *         first location within the array where the element is located
*/
```

Example: `int arr[] = {5, -3, 2, 5};` The method should return `0`.

```
public static int findMaximumIndex(int[] arr)
{
            /* some code */
}
```

I
```
int max = Integer.MIN_VALUE;
int loc = -1;
int i = 0;
while (i < arr.length)
{
  if (arr[i] > max)
  {
      max = arr[i];
      loc = i;
  }
  i++;
}
return loc;
```

II
```
int max = arr[arr.length - 1];
int loc = arr.length - 1;
int i = arr.length - 1;
while (i >= 0)
{
  if (arr[i] > max)
  {
      max = arr[i];
      loc = i;
  }
  i--;
}
return loc;
```

III
```
int i = 0;
int loc = 0;
int max = arr[loc];
while (i < arr.length)
{
  if (arr[i] > max)
  {
      max = arr[i];
      loc = i;
  }
  i++;
}
return loc;
```

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

GO ON TO THE NEXT PAGE.

17. The values of the static fields in the `Integer` class named `MIN_VALUE` and `MAX_VALUE` are constants containing the values `-2147483648` and `2147483647` respectively. Determine the statement(s) below that will throw a compiler error.

(A) `int min = Integer.MIN_VALUE;`
(B) `int min = Integer.MIN_VALUE - 1;`
(C) `int num = Integer.MIN_VALUE + Integer.MAX_VALUE;`
(D) `int max = 2147483648;`
(E) All of the statements above

18. What is the output after the following code is executed:

```
int num = 5;
System.out.print("" + half(num) + num);

public static double half(int n)
{
      return n/2;
}
```

(A) `2.05`
(B) `2.5`
(C) `7.0`
(D) `7.5`
(E) Nothing will print, as an error will be thrown.

GO ON TO THE NEXT PAGE.

19. Consider the following class declaration that is intended to represent a rectangle.

```
public class MyRectangle
{
   private int width;
   private int height;
   private int perimeter;

   MyRectangle(int w, int h)
   {
      width = w;
      height = h;
      int perimeter = 2 * (width + height);
   }

   public double getPerimeter()
   {
      return perimeter;
   }
}
```

What is the output after the following code is executed:

```
MyRectangle rect = new MyRectangle(2, 3);
System.out.print(rect.getPerimeter());
```

(A) `0`
(B) `0.0`
(C) `10`
(D) `10.0`
(E) None of the above; an error will be thrown.

GO ON TO THE NEXT PAGE.

Questions 20–22 refer to the following incomplete class declaration that is intended to represent a car.

```
public class Car
{
    private String model;
    private int numDoors;
    private boolean isFourWheelDrive;
    private int mpg;

    /**
     * Constructs a car
     */
    Car( )
    {

    }

    /**
     * Constructs a car
     * @param model
     * @param numDoors
     * @param isFourWheelDrive
     * @param mpg
     */
    Car(String model, int numDoors, boolean isFourWheelDrive, int mpg)
    {
        /* Implementation not shown */
    }

    /**
     * Compute the miles per gallon (milesDriven/gallons)
     * and stores the result in mpg, rounded to the nearest gallon
     * >= 0.5 would round up, < 0.5 would round down
     * @param milesDriven
     * @param gallons
    */
    public void setMpg(int milesDriven, double gallons)
    {
        /* Implementation not shown */
    }

    /**
     * Updates the model of the car
     * @param model
    */
    public void setModel(String model)
    {
       model = model;
    }
```

GO ON TO THE NEXT PAGE.

```
    /**
     *
     * @ return model
     */
    public String getModel()
    {
        return model;
    }
    /**
     * Updates the number of doors on the car
     * @param numDoors
    */
    public void setnumDoors(int numDoors)
    {
        /* Implementation not shown */
    }

    /**
     * Updates isFourWheelDrive true if the car has four-wheel drive, false if not
     * @param isFourWheelDrive
    */
    public void setIsFourWheelDrive (boolean isFourWheelDrive)
    {
        /* Implementation not shown */
    }


    /**
    * Returns the values stored in the object
    */
    public String toString()
    {
        return "Model: " + model + " is 4-wheel drive: " + isFourWheelDrive;
    }
}
```

20. The programmer wishes to add an additional constructor. Which of the following would be invalid as a constructor?

(A) `Car(String model)`
(B) `Car(String model, int mpg, boolean isFourWheelDrive)`
(C) `Car(String model, int numDoors)`
(D) `Car(String model, int mpg, boolean isFourWheelDrive, int numDoors)`
(E) `Car(String model, int numDoors, boolean isFourWheelDrive, int milesDriven, double gallons)`

GO ON TO THE NEXT PAGE.

21. The following code is in a client program.

```
Car fordTruck = new Car();

fordTruck.setModel("Tacoma");

if (fordTruck.getModel().equals("Tacoma"))
    fordTruck.setIsFourWheelDrive(true);

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

What will be output by the program?

(A) `Model: Tacoma is 4-wheel drive: true`
(B) `Model: Tacoma is 4-wheel drive: false`
(C) `Model: null is 4-wheel drive: true`
(D) `Model: null is 4-wheel drive: false`
(E) A `NullPointerException` will be thrown

22. Which of the following can replace /* Implementation not shown */ in the `setMpg` method?

(A) `mpg = milesDriven / gallons;`
(B) `mpg = milesDriven / gallons + 0.5;`
(C) `mpg = milesDriven / (int) gallons + 0.5;`
(D) `mpg = (int) (milesDriven / gallons) + 0.5;`
(E) `mpg = (int) (milesDriven / gallons + 0.5);`

GO ON TO THE NEXT PAGE.

Questions 23–25 refer to the following class declarations.

```
public class Bee
{
    private int lifeSpan;
    private String name;

    Bee(String n, int life)
    {
       lifeSpan = life;
       name = n;
    }

    public String getName()
    {
       return name;
    }

    public String toString()
    {
       return " The " + name + "s live " + lifeSpan + " months.";
    }
}

public class Queen extends Bee
{
    private int eggsPerDay;

    Queen(String name, int months, int eggs)
    {
       super(name, months);
       eggsPerDay = eggs;
    }

    public String toString()
    {
       return " The queen"  + " lays " + eggsPerDay + " eggs.";
    }
}
```

23. What is output after the following lines of code are executed in a client program?

```
Bee bee1 = new Queen("honey bee", 6, 2000);
System.out.println(bee1.toString());
```

(A) `The queen lays 2000 eggs.`
(B) `The honey bee lays 2000 eggs.`
(C) `The honey bees live 6 months.`
(D) `The honey bees live 6 months. The queen lays 2000 eggs.`
(E) An error will be thrown because of mismatched data types.

GO ON TO THE NEXT PAGE.

24. The programmer wishes to include an accessor method for the field called `name`. In which class should this method be included?

    (A) Only in the `Bee` class.
    (B) Only in the `Queen` class.
    (C) It could be included in either the `Bee` class or the `Queen` class.
    (D) It should be included in both classes.
    (E) The method should not be included in either class because the data is private.

25. Only queen bees lay eggs that hatch into productive bees for the colony. The other two types of bees in a colony are drone (male) and worker (female) bees. What information would be most helpful to determine the best design for a class implementation for drones and worker bees?

    (A) Whether the drones and workers have distinctly different lifespans than queens
    (B) Whether there are many more drones and workers than queens
    (C) Whether there are more drones than workers (or the reverse)
    (D) What data needs to be included to accurately represent the state and behavior of drones and workers
    (E) All the above

26. A program passes an array to a method in the same class to create multiples of 5. Determine the output of the following code.

```
double arr[];
arr = new double[5];
multOf5(arr);
for (int i = 0; i < arr.length; i++)
   System.out.print(arr[i] + "  ");

public static void multOf5(double a[])
{
   for (int i = 0; i < a.length; i++)
      a[i] = i * 5;
}
```

    (A) `0.0  5.0  10.0  15.0  20.0`
    (B) `0.0  0.0  0.0  0.0  0.0  0.0`
    (C) `0.0  5.0  10.0  15.0  20.0  25.0`
    (D) An error will occur because of a type mismatch.
    (E) An error will occur because the array was not initialized.

GO ON TO THE NEXT PAGE.

27. Consider the following method:

```
public static int[] op(int[][] matrix, int m)
{
       int[] result = new int[matrix.length];
       for (int j = 0; j < matrix.length; j++)
       {
          result[j] = matrix[m][j] - matrix[j][m];
       }
       return result;
}
```

The following code segment appears in the same class:

```
int mat[][] = {{1, 2, 3, 4}, {1, 3, 5, 7}, {2, 4, 6, 8}, {4, 3, 2, 1}};
int[] arr = op(mat, 3);
```

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

(A) `{0, -4, -6,  0}`
(B) `{0,  4,  6,  0}`
(C) `{8, 10, 10,  2}`
(D) `{2,  4,  6,  8}`
(E) `{3,  5,  6,  2}`

28. A programmer wishes to declare and initialize an `ArrayList` with random integers between 1 and 100. Choose the code that can replace /* missing code */ to accomplish the task.

```
ArrayList<Integer> list2= new ArrayList<Integer>();
for (int i = 0; i < 1000; i++)
{
    /* missing code */
}
```

(A) `list2.add((Math.random() * 100 + 1));`
(B) `list2.add((int)(Math.random() * 99 + 0.5));`
(C) `list2.add((int)(Math.random() * 100 + 0.5));`
(D) `list2.add((int)(Math.random() * 100 + 1));`
(E) `list2.add((Integer)(Math.random() * 100 + 0.5));`

GO ON TO THE NEXT PAGE.

29. What is printed after the following code is executed?

```
ArrayList<String> list1 = new ArrayList<String>();
list1.add("A");
list1.add("B");
list1.add("C");
list1.add("D");
list1.add("E");


for (int k = 0; k < list1.size(); k += 2)
{
      list1.remove(k);
}
for (int k = 1; k <= 3; k++)
{
      list1.add(1, "*");
}
for (String word : list1)
{
      System.out.print(word + " ");
}
```

(A) `B * * * C`
(B) `B * * * D`
(C) `B * * * E`
(D) `B * C * E`
(E) `B * * * C E`

30. What is the result of calling `mystery(5)`?

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

(A) `13`
(B) `15`
(C) `17`
(D) `19`
(E) `23`

GO ON TO THE NEXT PAGE.

31. What is the result of calling `mystery("PLANT");`

```
public static void mystery(String s)
{
   int i = 1;
   if (s.length() > 1)
   {
      String temp = s.substring(s.length() - i);
      System.out.println(temp);
      i++;
      mystery(temp);
   }
}
```

(A)
```
T
NT
ANT
LANT
PLANT
```
(B)
```
LANT
ANT
NT
T
```
(C)
```
LANT
ANT
NT
```
(D) `T`

(E) `PLANT`

GO ON TO THE NEXT PAGE.

Questions 32–33 refers to the following class declarations, which are intended to represent performances of plays.

```
public class Performance
{
    private String name;
    private String season;
    private int year;

    Performance(String n, String s, int y)
    {
       name = n;
       season = s;
       year = y;
    }
    public String toString()
    {
       return name + " will be performed in " + season + " of " + year;
    }
}

public class Play
{
    private Performance performance;
    private String mainCharacter;
    private String starringActor;

    Play(String n, String s, int y, String m, String star)
    {
       Performance p = new Performance(n, s, y);
       performance = p;
       mainCharacter = m;
       starringActor = star;
    }

    public String getMainCharacter()
    {
       return mainCharacter;
    }

    public String getStarringActor()
    {
       return starringActor;
    }
    public String toString()
    {
       return performance + " with " + starringActor + " as " + mainCharacter;
    }
}
```

GO ON TO THE NEXT PAGE.

32. What is the output after the following lines of code are executed in a client program?

```
Play p1 = new Play("Beauty and the Beast", "Winter",  2023, "Belle", "Kira");
Play p2 = new Play("Peter Pan", "Spring",  2023, "Peter", "Charlie");
Play p3 = new Play("Goldilocks and the Three Bears", "Summer",  2023,
                   "Goldilocks", "Sophia");
Play[] schedule = {p1, p2, p3};
System.out.println(schedule[1]);
```

(A) `Charlie as Peter`
(B) `Performance@7c53a9eb with Charlie as Peter`
(C) `Peter Pan will be performed in Spring of 2023`
(D) `Peter Pan will be performed in Spring of 2023 with Charlie as Peter`
(E) An error will be thrown.

33. Code is to be added to the same client program to build the following `ArrayList` of actors.

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

The `starringActor`(s) found in the schedule array (in the previous problem) will be stored in the `actors ArrayList`. Which of the following will properly add every the `starringActor` to the `ArrayList`?

(A)
```
for (Play p : schedule)
  actors.add(p.starringActor);
```
(B)
```
for (Play p : schedule)
  actors.add(p.getStarringActor());
```
(C)
```
for (Performance p : schedule)
  actors.add(p.getStarringActor());
```
(D)
```
for (int j = 0; j < schedule.length; j++)
  actors.add(schedule[j].starringActor);
```
(E)
```
for (int j = 0; j < schedule.length; j++)
  actors.add(schedule.get[j].getStarringActor());
```

GO ON TO THE NEXT PAGE.

34. Consider the following classes:

```
public class Flower
{
   private int height;

   public Flower()      /* constructor without parameters */
   {
      height = 0;
   }
   public String toString()
   {
      return "height = " +  height;
   }
}

public class Daffodil extends Flower
{
   /* accessor and mutator methods not shown */
}
```

Which of the following declarations is valid?

I. `Flower lily = new Flower();`
II. `Flower daffy = new Daffodil();`
III. `Daffodil daffo = new Daffodil();`

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

35. A binary search is used to find a target value in an array of 4000 elements, sorted in ascending order. Assuming a target value is in the array, what is the maximum number of searches that will occur to locate the target value?

(A) 11
(B) 12
(C) 15
(D) 40
(E) 2000

GO ON TO THE NEXT PAGE.

Questions 36–37 refer to the following sort method.

```
public static void sort(int arr[])
{
    int i;
    int num;
    int j;
    for (i = 1; i < arr.length; i++)      /* outer loop */
    {
       num = arr[i];
       j = i - 1;
       while (j >= 0 && arr[j] > num)     /* inner loop */
       {
              arr[j + 1] = arr[j];
              j = j - 1;
       }
       arr[j + 1] = num;
    }
}
```

36. If `sort` is called with an array of n elements, what is the maximum number of times the loop indicated by /* outer loop */ will be executed?

    (A) n
    (B) n / 2
    (C) n – 1
    (D) n + 1
    (E) 2^n

37. If `sort` is called with an array of n elements, on any given pass through /* outer loop */, what is the least number of times the loop indicated by /* inner loop */ will be executed?

    (A) 0
    (B) 1
    (C) n / 2
    (D) n – 2
    (E) n – 1

GO ON TO THE NEXT PAGE.

38. What changes to `mat` are implemented by the following code excerpt?

```
int for (int a = 0; a < mat.length; a++)
{
    temp = mat[a][0];
    for (int b = 1; b < mat[0].length; b++)
    {
       mat[a][b - 1] = mat[a][b];
    }
    mat[a][mat[0].length - 1] = temp;
}
```

(A) Every two columns are flipped. If there are an odd number of columns, there is no change to the last column.
(B) Every two rows are flipped. If there are an odd number of rows, there is no change to the last row.
(C) All columns are shifted left, and elements from the first column are moved to the last column.
(D) All rows are shifted upward, and elements from the first row are moved to the bottom row.
(E) An `outOfBounds` error is thrown.

GO ON TO THE NEXT PAGE.

Questions 39–40 refer to the following class declarations.

```
public class Sport
{
    private String season;
    private int year;

    Sport(String s, int y)
    {
       season = s;
       year = y;
    }

    public String toString()
    {
       return season + " " + year;
    }
}
public class Baseball extends Sport
{
    private int numPositions;

    Baseball(String s, int y)
    {
       super(s, y);
       numPositions = 10;
    }
}
```

39. The programmer wishes to store the following objects into an `ArrayList`.

```
Baseball b1 = new Baseball("Spring", 2022);
Baseball b2 = new Baseball("Summer", 2022);
Baseball b3 = new Baseball("Summer", 2023);
Baseball b4 = new Baseball("Summer", 2024);
Sport s1 = new Sport("Winter", 100);
```

Which of the following is a valid declaration for the `ArrayList`?

(A) `ArrayList<Sport> sports = new ArrayList<Sport>();`
(B) `ArrayList<Sport> sports = new ArrayList<Baseball>();`
(C) `ArrayList<Baseball> sports = new ArrayList<Sport>();`
(D) `ArrayList<Baseball> sports = new ArrayList<Baseball>();`
(E) These objects cannot be combined into a single `ArrayList`.

GO ON TO THE NEXT PAGE.

40. Which of the code excerpts will remove all objects from the `ArrayList` with a `season` designated as `"Summer"`?

I.
```
for (int i = 0; i < sports.size(); i++)
{
  if (sports.get(i).getSeason().equals("Summer"))
      sports.remove(i);
}
```

II.
```
int i = 0;
for (Sport a: sports)
{
if (a.getSeason().equals("Summer"))
     a.remove(i);
i++;
}
```

III.
```
for (int i = sports.size() - 1; i >= 0 ; i--)
{
  if (sports.get(i).getSeason().equals("Summer"))
      sports.remove(i);
}
```

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

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. This question involves writing methods for a `PigLatin` class. The `PigLatin` class stores a phrase in an instance variable and has three methods. The first will detect whether the parameter sent is a vowel, the second returns a word converted into pig latin, and the third will convert a phrase into pig latin.

```
public class PigLatin
{
    private String phrase;

    /** Constructs a new PigLatin object */
    public PigLatin(String phrase)
    {  this.phrase = phrase;  }

    /** Returns true if the letter sent is a vowel: a, e, i, o, or u
     *  If the letter is not a vowel returns false: "y" is not considered a vowel
     *  Preconditions: letter will not be null and will be exactly one character long,
     *                 letter will not be a space; letter will be lowercase
     *  Postconditions: letter is not modified,
     *                  only true or false, as explained above, is returned
     */
    public boolean isLetterAVowel(String letter)
    {  /* to be implemented in part (a) */  }


    /** Returns a word converted to pig latin
     *  Precondition: word.length > 0; all letters will be lowercase
     *                all words with word.length == 1 will be a single vowel
     *  Postcondition: the parameter word is not modified
     */
    public String convertWord(String word)
    {  /* to be implemented in part (b) */  }
```

GO ON TO THE NEXT PAGE.

```
    /** Returns the instance field phrase converted to pig latin
     *  Returns the empty string if the parameter word is empty or null
     *  Precondition: word may be null or any length
     *                the last character of the phrase will be a letter, not a space
     *                all characters will be lowercase
     *  Postcondition: phrase is not modified, returns phrase converted to pig latin
     */


public String convertPhrase()
    {   /* to be implemented in part (c) */    }

}
```

(a) Write the `PigLatin` method `isLetterAVowel`, which will return `true` if the parameter letter is a vowel (a, e, i, o, or u), and otherwise return `false`. The parameter letter will not be `null` or the empty string and will be exactly one character long. The value of the parameter will be a letter in lowercase.

Code segments	Value returned
`isLetterAVowel("a");`	`true`
`isLetterAVowel("e");`	`true`
`isLetterAVowel("u");`	`true`
`isLetterAVowel("b");`	`false`
`isLetterAVowel("y");`	`false`

Class information for this question

```
public class PigLatin

private String phrase;
public PigLatin(String phrase)
public boolean isLetterAVowel(String letter)
public String convertWord(String word)
public String convertPhrase()
```

GO ON TO THE NEXT PAGE.

The `PigLatin` class includes the method `isLetterAVowel`.

Complete method `isLetterAVowel` below.

```
/** Returns true if the letter sent is a vowel: a, e, i, o, or u
 *  If the letter is not a vowel returns false: "y" is not considered a vowel
 *  Preconditions: letter will not be null and will be exactly one character long,
 *                 letter will not be a space; letter will be lowercase
 *  Postconditions: letter is not modified
 *                  only true or false, as explained above, is returned
 */
public boolean isLetterAVowel(String letter)
```

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 `PigLatin` method `convertWord`, which will return a word converted to pig latin.

There will be three rules used to convert each word to pig latin:

- If `word` starts with a vowel, add the word "way" at the end of the word.
  "apple" would become "appleway", "a" would become "away"
- If `word` starts with a consonant and a vowel, move the consonant to the end of the word and add "ay".
  "dog" would become "ogday", "house" would become "ousehay"
- If `word` starts with two consonants, move both consonants to the end of the word and add "ay".
  "charge" would become "argechay", "sled" would become "edslay"

You must use `isLetterAVowel` appropriately to receive full credit.

Code segments	Value returned
`convertWord("ant");`	`"antway"`
`convertWord("cat");`	`"atcay"`
`convertWord("clone");`	`"oneclay"`

Class information for this question

```
public class PigLatin

private String phrase;
public PigLatin(String phrase)
public boolean isLetterAVowel(String letter)
public String convertWord(String word)
public String convertPhrase()
```

GO ON TO THE NEXT PAGE.

The `PigLatin` class includes the method `convertWord`.

Complete method `convertWord` below.

```
/** Returns a word converted to pig latin
 *  Precondition: word.length > 0; all letters will be lowercase
 *                all words with word.length == 1 will be a single vowel
 *  Postcondition: the parameter word is not modified
 *                 returns word converted to
 *                 pig latin
 */
public String convertWord(String word)
```

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) Write the `PigLatin` method `convertPhrase`, which will return the instance field `phrase` converted to pig latin. If `phrase` is empty or `null`, it will return the empty string. All characters in `phrase` will be lowercase letters.

You must use `convertWord` appropriately to receive full credit.

Code segments	Value returned
`convertPhrase("the cat is sleepy");`	`"ethay atcay isway eepyslay"`
`convertPhrase("where are you");`	`"erewhay areway ouyay"`
`convertPhrase(null);`	`""`
`convertPhrase("");`	`""`

Class information for this question

```
public class PigLatin

private String phrase;
public PigLatin(String phrase)
public boolean isLetterAVowel(String letter)
public String convertWord(String word)
public String convertPhrase()
```

GO ON TO THE NEXT PAGE.

The `PigLatin` class includes the method `convertPhrase`.

Complete method `convertPhrase` below.

```
/** Returns the instance field phrase converted to pig latin
 *  Returns the empty string if the parameter word is empty or null
 *  Precondition: word may be null or any length
 *                the last character of the phrase will be a letter, not a space
 *                all characters will be lowercase
 *  Postcondition: phrase is not modified, returns phrase converted to pig latin
 */
public String convertPhrase()
```

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.

This question involves writing a subclass for the `FrequentFlyerMember` class. This class holds information about frequent flyer members, including their names, account numbers, lifetime miles, and status levels.

2. The `FrequentFlyerMember` class definition is shown below.

```
public class FrequentFlyerMember
{
    private int acctNumber;
    private String flyerName;
    private int lifetimeMiles;
    private int statusLevel;

    FrequentFlyerMember(int n, String name, int miles)
    {
       acctNumber = n;
       flyerName = name;
       lifetimeMiles = miles;
       statusLevel = 1;
    }


    //* Adds miles to lifetimeMiles */
    public void addMiles(int miles)
    {
       lifetimeMiles += miles;
    }

    //* Changes the status level */
    public void setStatusLevel(int level)
    {
       statusLevel = level;
    }

    //* Prints selected status information */
    public String getStatusInfo()
    {
       return acctNumber + " " + flyerName + " level " + statusLevel;
    }

//*  other methods may exist but are not shown  */
}
```

You will write the class `PremierMember`, which is a subclass of `FrequentFlyerMember`.

A `PremierMember` has a `premierClubMembership` field represented by a boolean data type. It should be set to `true`. This flyer is also entitled to two `freeBags` which should be represented by an `int` field. The member will also have another String field named `otherFrequentFlyerMember` to store the name of any other frequent flyer memberships the flyer may be entitled to. These fields should be initialized in the constructor.

Information about the flyer's number, name, lifetime miles, and status level should be maintained and managed in the `FrequentFlyerMember` class.

GO ON TO THE NEXT PAGE.

Statement	Class Specifications or Print

`FrequentFlyerMember smith1 = new FrequentFlyerMember(14256, "Luke Smith", 1000);`	Class Specification:
	`smith1 is a FrequentFlyerMember`
	`number: 14256`
	`name: Luke Smith`
	`miles: 1000`
	`statusLevel: 1`

`smith1.addMiles(3025);`	Class Specification change:
	`smith1`
	`miles:3025`

`smith1.setStatusLevel(2);`	Class Specification change:
	`smith1`
	`statusLevel: 2`

`System.out.println(smith1.getStatusInfo());`	Printed: `14256 Luke Smith level 2`

`PremierMember jones1 = new PremierMember (97531, "Marcie Jones", "British Airways", 9000);`	Class Specification:
	`jones1 is a PremierMember`
	`number: 97531`
	`name: Marcie Jones`
	`miles: 9000`
	`statusLevel: 1`

`jones1.addMiles(10000);`	Class Specification change:
	`jones1`
	`miles: 19000`

`jones1.setStatusLevel(5);`	Class Specification change:
	`jones1`
	`statusLevel: 5`

`System.out.println(jones1.getStatusInfo());`	Printed: `97531 Marcie Jones level 5 also a member of British Airways`

GO ON TO THE NEXT PAGE.

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

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 `factorpair` is constructed from two (possibly non-distinct) numbers whose product is a given number. The `Factors` class constructs and stores all the given `FactorPair` objects of a number into an `ArrayList` with no duplicate pairs. The constructor uses a method: `buildArrayList` to create the `FactorPair` objects and create the `ArrayList`. The class utilizes a method `findMostPairs` to compare the number of `FactorPair` objects between two different numbers and return the number with the most `FactorPair` objects. If both numbers have the same number of `FactorPair` objects, the method returns `-1`. The `toString` method will print all the `FactorPair` objects within an `ArrayList`. You will write three methods of the `Factors` class: `buildArrayList`, `findMostPairs`, and `toString`.

```
public class FactorPair
{
    /** factor1 and factor2 represent two factors of a number */
    private int factor1;
    private int factor2;

    FactorPair(int f1, int f2)
    {
       factor1 = f1;
       factor2 = f2;
    }

    /* returns the first factor of a pair */
    public int getFactor1()
    { return factor1; }

    /* returns the second factor of a pair */
    public int getFactor2()
    { return factor2; }
}

public class Factors
{
    private int number;
    private ArrayList<FactorPair> pairs = new ArrayList<FactorPair>();

    Factors(int n)
    {
       number = n;
       pairs = buildArrayList(n);
    }

    /** Builds an ArrayList of all FactorPair objects of number
    /*  Precondition: n > 0
     *  Postcondition: the ArrayList will contain all FactorPair objects for number
     *                 the ArrayList will not contain duplicate FactorPair objects
     *                 return the ArrayList of FactorPair objects
     */
    public ArrayList<FactorPair> buildArrayList(int n)
    { /* to be implemented in part (a) }
```

GO ON TO THE NEXT PAGE.

```
/* Given two numbers as parameters, the method will return the
     * number with the most FactorPair objects
     * Precondition: n1 > 0, n2 > 0
     * Postcondition: The numbers are not modified
     *    return the number with the most FactorPair objects; if tied, -1 will be returned
     */
    public int findMostPairs(Factors f)
        {  /* to be implemented in part (b)  }


      /** Returns a string containing all the FactorPair objects in the ArrayList */

    public String toString()
        {  /* to be implemented in part (c) */  }

      /* other methods may be implemented but not shown */

}
```

GO ON TO THE NEXT PAGE.

(a) Write the `Factor` method `buildArrayList`, which will construct the `ArrayList` pairs for the number provided as the parameter.

Code segments	`FactorPairs` generated and added to the `ArrayList` `pairs`
`buildArrayList(24);`	pairs: `factor1: 1` `factor2: 24` | `factor1: 2` `factor2: 12` | `factor1: 3` `factor2: 8` | `factor1: 4` `factor2: 6`
`buildArrayList(45);`	pairs: `factor1: 1` `factor2: 45` | `factor1: 3` `factor2: 15` | `factor1: 5` `factor2: 9`
`buildArrayList(17);`	pairs: `factor1: 1` `factor2: 17`
`buildArrayList(20);`	pairs: `factor1: 1` `factor2: 20` | `factor1: 2` `factor2: 10` | `factor1: 4` `factor2: 5`

Class information for this question

```
public class Factors

private int number;
private ArrayList<FactorPair> pairs = new
ArrayList<FactorPair>();
Factors(int n)
public ArrayList<FactorPair> buildArrayList(int n)
public int findMostPairs(Factors f) {
public String toString() {

public class FactorPair

private int factor1;
private int factor2;
FactorPair(int f1, int f2)
public int getFactor1()
public int getFactor2()
```

GO ON TO THE NEXT PAGE.

The Factor class includes the method `buildArrayList`.

Complete method `buildArrayList` below.

```
/** Returns an ArrayList of all FactorPair objects of number
/*  Precondition:  n > 0
 *  Postcondition: the ArrayList will contain all FactorPair objects of n
 *                 the ArrayList will not contain duplicate FactorPair objects
 */
public ArrayList<FactorPair> buildArrayList(int n)
```

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 `Factor` class includes the method `findMostPairs`.

Given two numbers as parameters, the method will evaluate the number of `FactorPair` objects for each number and return the number that has the most `FactorPair` objects. If the number of `FactorPair` objects is the same for both numbers, `-1` will be returned.

You must use `buildArrayList` appropriately to receive full credit.

Code segments	Value returned (examples of the contents of the `ArrayList` are shown in part (a))

`Factors f1 = new Factors(20);`
 	
`Factors f2 = new Factors(24);`
 	
`System.out.println(f1.findMostPairs(f2));`
	`24`

`Factors f1 = new Factors(20);`
 	
`Factors f2 = new Factors(45);`
 	
`System.out.println(f1.findMostPairs(f2));`
	`-1`

`Factors f1 = new Factors(17);`
 	
`Factors f2 = new Factors(45);`
 	
`System.out.println(f1.findMostPairs(f2));`
	`45`

Class information for this question

```
public class Factors

private int number;
private ArrayList<FactorPair> pairs = new
ArrayList<FactorPair>();
Factors(int n)
public ArrayList<FactorPair> buildArrayList(int n)
public int findMostPairs(Factors f) {
public String toString() {

public class FactorPair
private int factor1;
private int factor2;
FactorPair(int f1, int f2)
public int getFactor1()
public int getFactor2()
```

GO ON TO THE NEXT PAGE.

Complete method `findMostPairs` below.

```
/* Returns the number with the most factored pairs; if tied, -1 will be returned
 * Precondition: n1 > 0, n2 > 0
 * Postcondition: The numbers are not modified
 */
public int findMostPairs(Factors f)
```

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 `Factor` class includes the method `toString`. This method will return a string containing containing the factors of each `FactorPair` object in the `ArrayList` using the format shown below.

Code segments	Value returned

`Factors f1 = new Factors(24);`
 	
`System.out.println(f1.toString());`
	`(1 24) (2 12) (3 8) (4 6)`

`Factors f2 = new Factors(17);`
 	
`System.out.println(f2.toString());`
	`(1 17)`

`Factors f3 = new Factors(20);`
 	
`System.out.println(f3.toString());`
	`(1 20) (2 10) (4 5)`

Class information for this question

```
public class Factors

private int number;
private ArrayList<FactorPair> pairs = new
ArrayList<FactorPair>();
Factors(int n)
public ArrayList<FactorPair> buildArrayList(int n)
public int findMostPairs(Factors f) {
public String toString() {

public class FactorPair

private int factor1;
private int factor2;
FactorPair(int f1, int f2)
public int getFactor1()
public int getFactor2()
```

GO ON TO THE NEXT PAGE.

Complete method `toString` below.

```
/* Returns a string containing all the FactorPair objects in the ArrayList pairs */
public String toString()
```

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. The `Array2DMultiples` class contains two static methods.

The first method, `buildMatrix`, is to construct and return a two-dimensional array. The parameters to the method will be a one-dimensional array and an `int` field `cols`. The length of the one-dimensional array will determine the number of rows in the two-dimensional array, while the `cols` field will determine the number of columns.

Each element in the parameter array will be used to determine the multiples that will populate the two-dimensional array. The `buildMatrix` method should work for any one-dimensional array that is provided, as well as any number of columns specified. For example, given the following parameters: `int [ ] arr = {7, 6, 2, 6}` and `cols = 5`, the two-dimensional array generated would have 4 rows and 5 columns. The first row would have multiples of 7, the second row would have multiples of 6, the third row would have multiples of 2, and the fourth row would have multiples of 6:

Multiples of 7	7	14	21	28	35
Multiples of 6	6	12	18	24	30
Multiples of 2	2	4	6	8	10
Multiples of 6	6	12	18	24	30

The second method, `eliminateDuplicateRows` will create a new array by removing any duplicate rows from the array. The above array would be changed to look like this:

Multiples of 7	7	14	21	28	35
Multiples of 6	6	12	18	24	30
Multiples of 2	2	4	6	8	10

```
/** Builds a two-dimensional array using the length of arr as the number of rows and
 * cols as the number of columns
 * @Precondition: arr.length > 0, cols > 0
 * @Postcondition: return the two-dimensional array
 */
public static int[][] buildMatrix(int [] arr, int cols)
{  /* to be implemented in part (a)  }

/** Create a new array by removing all duplicate rows from arrWithDups
 * @Postcondition: arrWithDups is not modified
 *                 returns the new two-dimensional array with no duplicate rows
 */
public static int [][] eliminateDuplicateRows(int [][] arrWithDups)
{  /* to be implemented in part (b)  }
```

GO ON TO THE NEXT PAGE.

(a) The `Array2DMultiples` class includes the method `buildMatrix`.

Code segments	Two-dimensional Matrix generated and returned
`int [] arr = {5, 2, 3, 5};` `int[][] arr2d = buildMatrix(arr, 6);`	5 10 15 20 25 30 2 4 6 8 10 12 3 6 9 12 15 18 5 10 15 20 25 30

Complete method `buildMatrix` below.

```
/** Builds a two-dimensional array using the length of arr as the number of rows and
 * cols as the number of columns
 * @Precondition: arr.length > 0, cols > 0
 * @Postcondition: return the two-dimensional array
 */
public static int[][] buildMatrix(int [] arr, int 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) The `Array2DMultiples` method includes the method `eliminateDuplicateRows`.

Examine the two-dimensional array parameter. Eliminate any duplicate rows and return a smaller two-dimensional array.

`arrWithDups:`

5	10	15	20	25	30
2	4	6	8	10	12
3	6	9	12	15	18
5	10	15	20	25	30

After the call: `eliminateDuplicateRows(arrWithDups)`, the array that is returned, will not have any duplicate rows.

5	10	15	20	25	30
2	4	6	8	10	12
3	6	9	12	15	18

GO ON TO THE NEXT PAGE.

Complete method `eliminateDuplicateRows` below.

```
/** Create a new array by removing all duplicate rows from arrWithDups
 * @Postcondition: arrWithDups is not modified
 *                 returns the new two-dimensional array with no duplicate rows
 */
    public static int [][] eliminateDuplicateRows(int [][] arrWithDups)
```

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