
AP Computer Science A: Practice Exam 2

Part I (Multiple Choice)

Time: 90 minutes

Number of questions: 40

Percent of total score: 50

Directions: Choose the best answer for each problem. Some problems take longer than others. Consider how much time you have left before spending too much time on any one problem.

Notes:

- You may assume all import statements have been included where they are needed.
- You may assume that the parameters in method calls are not null.
- You may assume that declarations of variables and methods appear within the context of an enclosing class.

1. Consider the following code segment.

```
for (int h = 5; h >= 0; h = h - 2)
{
    for (int k = 0; k < 3; k++)
    {
        if ((h + k) % 2 == 0)
            System.out.print((h + k) + " ");
    }
}
```

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

- (A) 5 3 1
- (B) 6 4 2
- (C) 51 31 11
- (D) 6 4 2 0
- (E) 51 31 11 00

2. Consider the following code segment.

```
List<String> foods = new ArrayList<String>();

foods.add("Hummus");
foods.add("Soup");
foods.add("Sushi");
foods.set(1, "Empanadas");
foods.add(0, "Salad");
foods.remove(1);
foods.add("Curry");
System.out.println(foods);
```

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

- (A) [Salad, Empanadas, Sushi, Curry]
- (B) [Hummus, Salad, Empanadas, Sushi, Curry]
- (C) [Hummus, Soup, Sushi, Empanadas, Salad, Curry]
- (D) [Soup, Empanadas, Sushi, Curry]
- (E) [Hummus, Sushi, Salad, Curry]

3. Consider the following output.

A A A A P P P P C C C C

Which of the following code segments will produce this output?

```

I. String[] letters = {"A", "P", "C", "S"};
   for (int i = letters.length; i > 0; i--)
   {
       if (!letters[i].equals("S"))
       {
           for (int k = 0; k < 4; k++)
               System.out.print(letters[k] + " ");
       }
   }

II. String letters = "APCS";
    for (int i = 0; i < letters.length(); i++)
    {
        String s = letters.substring(i, i + 1);
        for (int num = 0; num < 4; num++)
        {
            if (!s.equals("S"))
                System.out.print(s + " ");
        }
    }

III. String[][] letters = { {"A", "P"}, {"C", "S"} };
    for (String[] row : letters)
    {
        for (String letter : row)
        {
            if (!letter.equals("S"))
                System.out.print(letter + " ");
        }
    }

```

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

Questions 4–5 refer to the following two classes.

```

public abstract class Vehicle
{
    private int fuel;

    public Vehicle(int fuelAmt)
    {
        fuel = fuelAmt;
    }

    public void start()
    {
        System.out.println("Vroom");
    }

    public void changeFuel(int change)
    {
        fuel = fuel + change;
    }

    public int getFuel()
    {
        return fuel;
    }

    public abstract void useFuel();
}

public class Boat extends Vehicle
{
    public Boat(int fuelAmount)
    {
        super(fuelAmount);
    }

    public void useFuel()
    {
        super.changeFuel(-2);
    }

    public void start()
    {
        super.start();
        useFuel();
        System.out.println("Remaining Fuel " + getFuel());
    }
}

```

4. Assume the following declaration appears in a client program.

```
Boat yacht = new Boat(20);
```

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

- (A) Vroom
- (B) Remaining Fuel 18
- (C) Remaining Fuel 20
- (D) Vroom
Remaining Fuel 18
- (E) Vroom
Remaining Fuel 20

5. Which of the following statements results in a compile-time error?

- I. `Boat sailboat = new Boat(2);`
- II. `Vehicle tanker = new Boat(20);`
- III. `Vehicle car = new Vehicle(10);`

- (A) I only
- (B) II only
- (C) III only
- (D) II and III only
- (E) None of these options will result in a compile-time error.

6. Consider the following recursive method.

```
public int weird(int num)
{
    if (num <= 0)
        return num;
    return weird(num - 2) + weird(num - 1) + weird(num);
}
```

What value is returned as a result of the call `weird(3)`?

- (A) -2
- (B) 3
- (C) 4
- (D) 6
- (E) Nothing is returned. Infinite recursion causes a stack overflow error.

7. Consider the following method.

```

public boolean verifyValues(int[] values)
{
    for (int index = values.length - 1; index > 0; index--)
    {
        /* missing code */
        return false;
    }
    return true;
}

```

The method `verifyValues` is intended to return `true` if the array passed as a parameter is in ascending order (least to greatest), and `false` otherwise.

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

- (A) `if (values[index] <= values[index - 1])`
- (B) `if (values[index + 1] < values[index])`
- (C) `if (values[index] >= values[index - 1])`
- (D) `if (values[index - 1] > values[index])`
- (E) `if (values[index] < values[index + 1])`

8. Consider the following code segment.

```

int[][] matrix = new int[3][3];
int value = 0;

for (int row = 0; row < matrix.length; row++)
{
    for (int column = 0; column < matrix[row].length; column++)
    {
        matrix[row][column] = value;
        value++;
    }
}

int sum = 0;
for (int[] row : matrix)
{
    for (int number : row)
        sum = sum + number;
}

```

What is the value of `sum` after the code segment has been executed?

- (A) 0
- (B) 9
- (C) 21

(D) 36

(E) 72

9. Consider the following class used to represent a student.

```
public class Student
{
    private String name;
    private int year;

    public Student()
    {
        name = "name";
        year = 0;
    }

    public Student(String myName, int myYear)
    {
        name = myName;
        year = myYear;
    }
}
```

Consider the ExchangeStudent class that extends the Student class.

```
public class ExchangeStudent extends Student
{
    private String country;
    private String language;

    public ExchangeStudent(String myName, int myYear,
                           String myCountry, String myLanguage)
    {
        super(myName, myYear);
        country = myCountry;
        language = myLanguage;
    }
}
```

Which of the following constructors could also be included in the ExchangeStudent class without generating a compile-time error?

I.

```
public ExchangeStudent(String myName, int myYear, String myCountry)
{
    super(myName, myYear);
    country = myCountry;
    language = "English";
}
```

```
II. public ExchangeStudent(String myCountry, String myLanguage)
{
    super("name", "2015");
    country = myCountry;
    language = myLanguage;
}
```

```
III. public ExchangeStudent()
{
}
```

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

10. Consider the following code segment.

```
int number = Integer.MAX_VALUE;
while (number > 0)
{
    number = number / 2;
}

for (int i = number; i > 0; i++)
{
    System.out.print(i + " ");
}
```

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

- (A) 0
- (B) 1
- (C) 1073741823
- (D) Nothing will be printed. The code segment will terminate without error.
- (E) Nothing will be printed. The first loop is an infinite loop.

11. Consider the following method.


```

/** Precondition: numbers.size() > 0
 */
public int totalValue(List<Integer> numbers)
{
    int total = 0;
    for (Integer val : numbers)
    {
        if (val > 1 && numbers.size() - 3 > val)
            total += val;
    }
    return total;
}

```

Assume that the ArrayList passed as a parameter contains the following Integer values.

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

What value is returned by the call totalValue?

- (A) 20
- (B) 21
- (C) 27
- (D) 44
- (E) 45

Questions 12–13 refer to the following interface and class definitions, which are components of a video game.

```

public interface Ship
{
    /** The move method changes the location of the Ship
     */
    void move();
}

```

A PlayerShip object is controlled by the human player.

```

public class PlayerShip implements Ship
{
    public void move()
    { /* implementation not shown */ }

    /** @return the current y position of the PlayerShip
     */
    public int getYPosition()
    { /* implementation not shown */ }

    /* Additional implementation not shown */
}

```

An EnemyShip object is controlled by the computer.

```
public class EnemyShip implements Ship
{
    public int xPosPosition;
    public int yPosPosition;

    public void move()
    { /* missing code */ }

    /* Additional implmentation not shown */
}
```

12. An EnemyShip object has the following behavior:

- An EnemyShip always moves from right to left in the x direction by subtracting 1 from the xPosPosition for each move. When xPosPosition reaches 0, it is reset to 1000.
- An EnemyShip always moves toward a PlayerShip in the y direction, or stays in its current y position if the PlayerShip and the EnemyShip have the same yPosPosition (since the EnemyShip cannot get any closer).

Given PlayerShip player, which of the following is a correct implementation of the move method in the Enemyship class?

I.

```
public void move()
{
    if (xPosPosition == 0)
        xPosPosition = 1000;
    else
        xPosPosition = xPosPosition - 1;

    if (player.getYPosition() > 0)
        yPosPosition++;
}
```

II.

```
public void move()
{
    if (xPosPosition == 0)
        xPosPosition = 1000;
    else
        xPosPosition -= 1;

    if (player.getYPosition() > yPosPosition)
        yPosPosition++;
    else
        yPosPosition--;
}
```

```

III. public void move()
{
    if (xPosition == 0)
        xPosition = 1000;
    else
        xPosition = xPosition - 1;

    if (player.getYPosition() > yPosition)
        yPosition++;
    else if (player.getYPosition() < yPosition)
        yPosition--;
}

```

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

13. Which of the following classes correctly implements the Ship interface?

```

I. public class BattleShip implements Ship
{
    private int x;
    private int y;

    public Ship(int myX, int myY)
    {
        x = myX;
        y = myY;
    }

    public boolean move()
    {
        return x + y > 0;
    }
}

II. public abstract class Zoomer implements Ship
{
    public abstract void move();
}

```

```

III. public class Sailboat extends Ship
{
    private int x;
    private int y;
    public Sailboat()
    {
        x = 0;
        y = 1000;
    }

    public void move()
    {
        x++;
        y--;
    }
}

```

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

14. A bank will approve a loan for any customer who fulfills one or more of the following requirements:

- Has a credit score ≥ 640
- Has a cosigner for the loan
- Has a credit score ≥ 590 and has collateral

Which of the following methods will properly evaluate the customer's eligibility for a loan?

```

I. public boolean canGetLoan(int credit, boolean cosigner, boolean coll)
{
    if (credit >= 640 || cosigner || credit >= 590 && coll)
        return true;
    return false;
}

II. public boolean canGetloan(int credit, boolean cosigner, boolean coll)
{
    if (!credit >= 640 || !cosigner || !(credit >= 590 && coll))
        return false;
    else
        return true;
}

```

```

III. public boolean canGetLoan(int credit, boolean cosigner, boolean coll)
    {
        if (coll && credit >= 590)
            return true;
        if (credit >= 640)
            return true;
        if (cosigner)
            return true;
        return false;
    }

```

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

15. Consider the following code segment.

```

int value = 33;
boolean calculate = true;
while (value > 5 || calculate)
{
    if (value % 3 == 0)
        value = value - 2;
    if (value / 4 < 3)
        calculate = false;
}
System.out.print(value);

```

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

- (A) 0
- (B) 1
- (C) 8
- (D) 33
- (E) Nothing will be printed. It is an infinite loop.

16. Assume that planets has been correctly instantiated and initialized to contain the names of the planets. Which of the following code segments will reverse the order of the elements in planets?

```

(A) for (int index = planets.length / 2; index >= 0; index--)
    {
        String temp = planets[index];
        planets[index] = planets[index + 1];
        planets[index + 1] = temp;
    }

```

- (B)

```
int index = planets.length / 2;
while (index >= 0)
{
    String s = planets[index];
    planets[index] = planets[index + 1];
    index--;
}
```
- (C)

```
String[] newPlanets = new String[planets.length];
for (int i = planets.length - 1; i >= 0; i--)
{
    newPlanets[i] = planets[i];
}
planets = newPlanets;
```
- (D)

```
for (int index = 0; index < planets.length; index++)
{
    String temp = planets[index];
    planets[index] = planets[planets.length - 1 - index];
    planets[planets.length - 1 - index] = temp;
}
```
- (E)

```
for (int index = 0; index < planets.length / 2; index++)
{
    String temp = planets[index];
    planets[index] = planets[planets.length - 1 - index];
    planets[planets.length - 1 - index] = temp;
}
```

17. When designing a class hierarchy, what should be true of an abstract class?

- (A) An abstract class should contain the methods and class variables that all of its subclasses have in common, even if the correct implementations of the methods are unknown.
- (B) An abstract class must be the largest most complicated superclass for which all other classes should derive.
- (C) An abstract class should contain all final variables and static methods of the hierarchy.
- (D) An abstract class should contain the specific implementations of every method inherited from an interface that the abstract class implements.
- (E) An abstract class should contain basic implementation of all methods so that subclasses can override its implementation with more specific and correct methods.

18. Consider the following code segment.

```
int num = (int)(Math.random() * 30 + 20);
num += 5;
num = num / 5;
System.out.print(num);
```

What are the possible values that could be printed to the console?

- (A) All real numbers from 4 to 10 (not including 10)
- (B) All integers from 5 to 10 (inclusive)
- (C) All integers from 20 to 49 (inclusive)
- (D) All integers from 25 to 54 (inclusive)
- (E) All real numbers from 30 to 50 (inclusive)

19. Consider the following code segment.

```
String exampleString = "computer";
List<String> words = new ArrayList<String>();
for (int k = 0; k < exampleString.length(); k++)
{
    words.add(exampleString.substring(k));
    k++;
}
```

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

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

- (A) [computer, mputer, uter, er]
- (B) [computer, comput, comp, co]
- (C) [computer, computer, computer, computer]
- (D) [computer, omputer, mputer, puter, uter, ter, er, r]
- (E) Nothing is printed. There is an ArrayListIndexOutOfBoundsException.

20. Consider the following incomplete method.

```
public boolean validation(int x, int y)
{
    /* missing code */
}
```

The following table shows several examples of the desired result of a call to validation.

x	y	Result
1	0	true
3	6	true
4	1	true
1	3	false
2	6	false
5	5	false

Which of the following code segments should replace */* missing code */* to

produce the desired return values?

- (A) return `x > y`;
- (B) return `(x % y) > 1`
- (C) return `(x + y) % 2 == 0`
- (D) return `(x + y) % x == y`;
- (E) return `(x + y) % 2 > (y + y) % 2`;

21. Consider the following method that is intended to remove all Strings from words that are less than six letters long.

```
private void letterCountCheck(ArrayList<String> words)
{
Line 1:     int numWord = 0;
Line 2:     while (numWord <= words.size())
            {
Line 3:         if (words.get(numWord).length() < 6)
            {
Line 4:             words.remove(numWord);
            }
            else
            {
Line 5:             numWord++;
            }
        }
    }
```

Which line of code contains an error that prevents letterCountCheck from working as intended?

- (A) Line 1
 - (B) Line 2
 - (C) Line 3
 - (D) Line 4
 - (E) Line 5
22. Consider the following recursive method.

```
public void wackyOutput(String wacky)
{
    if (wacky.length() < 1)
        return;
    wacky = wacky.substring(1, wacky.length());
    wackyOutput(wacky);
    System.out.print(wacky);
}
```

What is printed as a result of executing the call `wackyOutput("APCS")`?

- (A) PC
- (B) SCPA
- (C) SCSPCS
- (D) PCSCSS
- (E) SCSPCSAPCS

23. Consider the following method.

```
public String calculate(int num1, int num2)
{
    if (num1 >= 0 && num2 >= 0)
        return "Numbers are valid";
    return "Numbers are not valid";
}
```

Which of the following methods will give the exact same results as calculate?

- I.

```
public String calculate1(int num1, int num2)
{
    if (!(num1 < 0 || num2 < 0))
        return "Numbers are valid";
    else
        return "Numbers are not valid";
}
```
- II.

```
public String calculate2(int num1, int num2)
{
    if (num1 < 0)
        return "Numbers are not valid";
    if (num2 < 0)
        return "Numbers are not valid";
    else
        return "Numbers are valid";
}
```
- III.

```
public String calculate3(int num1, int num2)
{
    if (num1 + num2 < 0)
        return "Numbers are not valid";
    return "Numbers are valid";
}
```

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

24. Consider the following class declaration.

```

public class TestClass
{
    private double value;

    public TestClass(double myValue)
    {
        value = myValue;
    }

    public void addValue(double add)
    {
        value += add;
    }

    public void reduceValue(double reduce)
    {
        value -= reduce;
    }

    public double getValue()
    {
        return value;
    }
}

```

The following code segment is executed in the main method.

```

TestClass test1 = new TestClass(9.0);
TestClass test2 = new TestClass(17.5);
test1.reduceValue(3.0);
test2.addValue(1.5);
test2 = test1;
test2.reduceValue(6.0);
System.out.print(test2.getValue() + test1.getValue());

```

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

- (A) 0.0
- (B) 3.0
- (C) 12.0
- (D) 19.0
- (E) 27.5

25. Assume `truth1` and `truth2` are boolean variables that have been properly declared and initialized.

Consider this expression.

```
(truth1 && truth2) || ((!truth1) && (!truth2))
```

Which expression below is its logical equivalent?

- (A) `truth1 != truth2`
- (B) `truth1 || truth2`
- (C) `truth1 && truth2`
- (D) `!truth1 && !truth2`
- (E) `truth1 == truth2`

26. The following insertion sort method sorts the array in increasing order. This method includes an additional `System.out.print(value + " ")` statement that is not normally included in the insertion sort.

```
// postcondition: reals is sorted in increasing order
public void sortIt(double[] reals)
{
    for (int i = 1; i < reals.length; i++)
    {
        double value = reals[i];

        int j;
        for (j = i - 1; j >= 0 && reals[j] > value; j--)
            reals[j + 1] = reals[j];
        reals[j + 1] = value;

        System.out.print(value + " "); //Additional statement
    }
}
```

Consider the following array.

```
double[] costs = {2.6, 3.1, 7.8, 1.4, 6.6, 5.2, 9.1};
```

What is printed as a result of executing the call `sortIt(costs)`?

- (A) 3.1 7.8 1.4 6.6 5.2
- (B) 3.1 7.8 1.4 6.6 5.2 9.1
- (C) 2.6 3.1 7.8 1.4 6.6 5.2 9.1
- (D) 1.4 2.6 3.1 5.2 6.6 7.8 9.1
- (E) 3.1 3.1 7.8 7.8 1.4 1.4 6.6 6.6 5.2 5.2 9.1 9.1

27. Consider the following interface.

```
public interface Polygon
{
    /** returns true if the ordered pair is inside the polygon
     *
     *         false otherwise
     */
    boolean contains(int x, int y);
}
```

Consider the following class.

```
public class Rectangle implements Polygon
{
    private int topLeftX, topLeftY, bottomRightX, bottomRightY;

    /**
     * Precondition: topLeftX < bottomRightX
     * Precondition: topLeftY < bottomRightY
     * The y value increases as it moves down the screen.
     * @param topLeftX The x value of the top left corner
     * @param topLeftY The y value of the top left corner
     * @param bottomRightX The x value of the bottom right corner
     * @param bottomRightY The y value of the bottom right corner
     */
    public Rectangle(int topLX, int topLY, int bottomRX, int bottomRY)
    {
        topLeftX = topLX;
        topLeftY = topLY;
        bottomRightX = bottomRX;
        bottomRightY = bottomRY;
    }
}
```

Which of the following is a correct implementation of the contains method?

- (A)

```
public boolean contains(int x, int y)
{
    return x > topLeftX && y > topLeftY &&
           x < bottomRightX && y < bottomRightY;
}
```
- (B)

```
public boolean contains(int x, int y)
{
    if (topLeftX > x)
        return true;
    else if (topLeftY > y)
        return true;
    else if (bottomRightX < x)
        return true;
    else if (bottomRightY < y)
        return true;
    return false;
}
```
- (C)

```
public boolean contains(int x, int y)
{ return super.contains(x, y); }
```
- (D)

```
public boolean contains(int x, int y)
{ return Polygon.contains(x, y); }
```

```
(E) public boolean contains(int x, int y)
{
    boolean result = false;
    if (topLeftX < x && topLeftY < y)
        result = true;
    if (bottomRightX > x && bottomRightY > y)
        result = true;
    return result;
}
```

28. Consider the following method.

```
public ArrayList<String> rearrange(ArrayList<String> myList)
{
    for (int i = myList.size() / 2; i >= 0; i--)
    {
        String wordA = myList.remove(i);
        myList.add(wordA);
    }
    return myList;
}
```

Assume that `ArrayList<String> list` has been correctly instantiated and populated with the following entries.

```
["Nora", "Charles", "Madeline", "Nate", "Silja", "Garrett"]
```

What are the values of the `ArrayList` returned by the call `rearrange(list)`?

- (A) ["Silja", "Garrett", "Nate", "Madeline", "Charles", "Nora"]
- (B) ["Madeline", "Charles", "Nora", "Nate", "Silja", "Garrett"]
- (C) ["Nate", "Silja", "Garrett", "Nora", "Charles", "Madeline"]
- (D) ["Nora", "Charles", "Madeline", "Nate", "Silja", "Garrett"]
- (E) Nothing is returned. `ArrayListIndexOutOfBoundsException`

29. Consider the following code segment.

```
int varA = -30;
int varB = 30;
while (varA != 0 || varB > 0)
{
    varA = varA + 2;
    varB = Math.abs(varA + 2);
    varA++;
    varB = varB - 5;
}
System.out.println(varA + "    " + varB);
```

What will be printed as a result of executing the code segment?

- (A) -30 30
- (B) -4 0

- (C) 0 -4
- (D) 0 0
- (E) Nothing will be printed. It is an infinite loop.

30. Consider the following code segment.

```
int[][] grid = { {0, 1, 2},
                 {3, 4, 5},
                 {6, 7, 8},
                 {9, 10, 11} };

int[][] newGrid = new int[grid[0].length][grid.length];

for (int row = 0; row < grid.length; row++)
{
    for (int col = 0; col < grid[row].length; col++)
    {
        newGrid[col][row] = grid[row][col];
    }
}
```

What is the value of newGrid[2][1] as a result of executing the code segment?

- (A) 3
- (B) 4
- (C) 5
- (D) 7
- (E) Nothing is printed. There is an ArrayIndexOutOfBoundsException.

31. Consider the following code segment.

```
int[][] grid = new int[3][3];
for (int row = 0; row < grid.length; row++)
{
    for (int col = 0; col < grid[row].length; col++)
    {
        grid[row][col] = 1;
        grid[col][row] = 2;
        grid[row][row] = 3;
    }
}
```

Which of the following shows the values in grid after executing the code segment?

- (A) { {1, 1, 1},
 {1, 1, 1},
 {1, 1, 1} };

- (B) { {2, 2, 2},
 {2, 2, 2},
 {2, 2, 2} };
- (C) { {3, 2, 2},
 {2, 3, 2},
 {2, 2, 3} };
- (D) { {3, 1, 1},
 {1, 3, 1},
 {1, 1, 3} };
- (E) { {3, 2, 2},
 {1, 3, 2},
 {1, 1, 3} };

32. What can the following method best be described as?

```
public int mystery(int[] array, int a)
{
    for (int i = 0; i < array.length; i++)
    {
        if (array[i] == a)
            return i;
    }
    return -1;
}
```

- (A) Insertion sort
- (B) Selection sort
- (C) Binary search
- (D) Merge sort
- (E) Sequential search

Questions 33–36 refer to the following classes.

```

public abstract class Soda implements Beverage
{
    private String size;
    private double price;

    public Soda()
    { /* implementation not shown */ }

    public Soda(String size, double price)
    { /* implementation not shown */ }

    public Soda(String size)
    { /* implementation not shown */ }

    /* no other constructors */

    public abstract int getCalories(String size);

    public boolean isHot()
    { return false; }

    public abstract boolean hasCaffeine();

    /* Additional implementation not shown */
}

```



```

public class Coffee implements Beverage
{
    private double price;

    public Coffee()
    { /* implementation not shown */ }

    public Coffee(double price)
    { /* implementation not shown */ }

    /* no other constructors */

    public boolean isHot()
    { return true; }

    public boolean hasCaffeine()
    { return true; }

    /* no other methods */
}

public class RootBeer extends Soda
{
    public RootBeer(String size, double price)
    {
        super(size, price);
    }

    /* no other constructors */

    /* Additional implementation not shown */
}

```

33. Which of the following will compile without error?

- I. Beverage yum1 = new Coffee();
- II. Soda yum2 = new RootBeer("large", 3.00);
- III. Beverage yum3 = new Soda("small");

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

34. Which of the following must be included in the implementation of the RootBeer

class?

- I. `public RootBeer()`
- II. `public boolean hasCaffeine()`
- III. `public boolean isHot()`
- IV. `public int getCalories()`

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

35. Which of the following methods could be part of the Beverage interface?

- I. `boolean hasCaffeine()`
- II. `boolean isHot()`
- III. `int getCalories()`

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

36. Which of the following code segments will compile without error?

- (A)

```
ArrayList<Beverage> drinks = new ArrayList<Beverage>();  
for (Beverage b : drinks)  
    System.out.println(b.hasCaffeine());
```
- (B)

```
ArrayList<RootBeer> drinks = new ArrayList<Soda>();  
for (Beverage b : drinks)  
    System.out.println(b.getCalories());
```
- (C)

```
ArrayList<Soda> drinks = new ArrayList<Beverage>();  
for (Beverage b : drinks)  
    System.out.println(b.getCalories());
```
- (D)

```
ArrayList<Beverage> drinks = new ArrayList<Coffee>();  
for (Beverage b : drinks)  
    System.out.println(b.getCalories());
```
- (E)

```
ArrayList<Soda> drinks = new ArrayList<Coffee>();  
for (Beverage b : drinks)  
    System.out.println(b.getCalories());
```

37. Consider the following method.

```

public int countLetters(String word, String letter)
{
    int count = 0;
    for (int index = 0; index < word.length(); index++)
    {
        if ( /* missing code */ )
            count++;
    }
    return count;
}

```

What could replace `/* missing code */` to allow the method to return the number of times letter appears in word?

- (A) `word.substring(index).equals(letter)`
- (B) `word.substring(index, index + 1) == letter`
- (C) `word.indexOf(letter) == letter.indexOf(letter)`
- (D) `word.substring(index, index + 1).equals(letter)`
- (E) `letter.equals(word.substring(index).indexOf(letter))`

- 38.** Assume `obscureAnimals` is an `ArrayList<String>` that has been correctly constructed and populated with the following items.

```
["okapi", "aye-aye", "cassowary", "echidna", "sugar glider", "jerboa"]
```

Consider the following code segment.

```

for (int i = 0; i < obscureAnimals.size(); i++)
{
    if (obscureAnimals.get(i).compareTo("pink fairy armadillo") < 0)
        obscureAnimals.remove(i);
}
System.out.print(animals);

```

What will be printed as a result of executing the code segment?

- (A) `[]`
- (B) `[sugar glider]`
- (C) `[aye-aye, echidna, sugar glider]`
- (D) `[aye-aye, echidna, sugar glider, jerboa]`
- (E) Nothing will be printed. There is an `ArrayListIndexOutOfBoundsException`.

Questions 39–40 refer to the following classes.

Consider the following class declarations.

```

public class Building
{
    int sqFeet;
    int numRooms;

    public Building(int ft, int rms)
    {
        sqFeet = ft;
        numRooms = rms;
    }

    public int getSqfeet()
    { return sqFeet; }

    public String getSize()
    {
        return numRooms + " rooms and " + sqFeet + " square feet";
    }

    /* Additional implementation not shown */
}

public class House extends Building
{
    int numBedRooms;
    public House(int ft, int rms, int bedrms)
    {
        super(ft, rms);
        numBedRooms = bedrms;
    }

    public String getSize()
    {
        return super.getSqFeet() + " square feet and " + numBedRooms + " bedrooms.";
    }
}

```

- 39.** Assume that `ArrayList<Building> list` has been correctly instantiated and populated with `Building` objects.

Which of the following code segments will result in the square feet in each building being printed?

- I. `for (int i = 0; i < list.size(); i++)`
 `System.out.println(list.getSqFeet(i));`
- II. `for (int i = 0; i < list.size(); i++)`
 `System.out.println(list[i].getSqFeet());`

```

III. for (int i = 0; i < list.size(); i++)
      System.out.println(list.get(i).getSqFeet());
IV.  for (Building b : list)
      System.out.println(list.getSqFeet());
V.   for (Building b : list)
      System.out.println(b.getSqFeet());

```

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

40. Consider the following code segment.

```

Building b1 = new Building(2000, 3);
Building b2 = new House(2500, 8, 4);
Building b3 = b2;
Building[] buildings = new Building[3];
buildings[0] = b1;
buildings[1] = b2;
buildings[2] = b3;

```

What will be printed by the following code segment?

```

for (int i = 0; i < buildings.length; i++)
    System.out.println(buildings[i].getSize());

```

- (A) 3 rooms and 2000 square feet
8 rooms and 2500 square feet
8 rooms and 2500 square feet
- (B) 3 rooms and 2000 square feet
2500 square feet and 4 bedrooms
2500 square feet and 4 bedrooms
- (C) 2000 square feet and 3 bedrooms
2500 square feet and 4 bedrooms
2500 square feet and 4 bedrooms
- (D) 3 rooms and 2000 square feet
2500 square feet and 4 bedrooms
3 rooms and 2000 square feet
- (E) There is an error. Nothing will be printed.

STOP. End of Part I.