

Circle your answers & fill out the Scantron form if provided. You may use the Java Quick Reference at http://apcentral.collegeboard.com/apc/public/repository/ap_comp_sci_a_quick_reference.pdf

Part I - Multiple Choice

1. Consider the following method:

```
public static double solve(int n)
{
    double answer = 1.0;
    for (int k = 1; k < n; k++)
        answer *= (double) k/(k + 1);
    return answer;
}
```

What value is returned by a call to method solve?

- a. 0.0
- b. the kth root of n
- c. 1/n
- d. 1/(n - 1)
- e. $(1/2)^n$

2. Consider the following incomplete class declaration:

```
public class SeatLocation
{
    private String row; // "A", "B", ..., "Z", "AA", "BB", ... , "ZZ"
    private int seatNumber;

    public String getRow() { . . . }
    public int getSeatNumber() { . . . }
    // constructor and other methods not shown
}
```

```
public class Ticket
{
    private double price;
    private SeatLocation loc;

    public void setPrice(double p) { price = p; }
    public void setSeatLocation(SeatLocation s) { loc = s; }
    public double getPrice() { return price; }
    public SeatLocation getSeatLocation() { return loc; }
    // constructor and other methods not shown
}
```

Assume the following valid declarations have been made:

```
SeatLocation s = new SeatLocation("Q", 107);
Ticket t = new Ticket(85, 00, s);
double totalCost = 0.0;
```

Which of the following statements in a client program is valid?

- a. `if (s.row = "H") System.out.println("Too far back.");`
- b. `t.setSeatLocation("E", 19);`
- c. `System.out.println(t.getSeatLocation.getRow(), s.getSeatNumber());`
- d. `t.setPrice(0.5 * t.getPrice());`
- e. `for (String row = "A"; row.compareTo("Z") < 0; row++)
 totalCost += t.getPrice(row, loc.getSeatNumber());`

3. Consider the following code segment:

```
int x = 2;  
int y = 4;  
x += y;  
y += x;  
System.out.println("x = " + x + ", y = " + y);
```

What output is produced by executing this code?

- a. `x = 2, y = 4`
- b. `x = 6, y = 6`
- c. `x = 6, y = 10`
- d. `x = 6, y = 12`
- e. `x = 24, y = 42`

4. Consider the following method:

```
public boolean evaluate(boolean p, boolean q)  
{  
    return ((p && q) || !(p || q));  
}
```

What is returned by evaluate?

- a. evaluate always returns false.
- b. evaluate returns true when p and q are both true or both false.
- c. evaluate returns true only when p and q are both true.
- d. evaluate returns true only when p and q are both false.
- e. evaluate returns true when either p or q is true and the other is false.

5. Consider the following method, which is intended to count the number of times a value is doubled before it is at least as large as a target value:

```
// precondition: d > 0  
public void doubleUp(double d)  
{  
    static final double TARGET = 100.0;  
    int count;  
    while (d < TARGET)  
    {  
        count = 0;  
        d *= 2;  
        count++;  
    }  
    System.out.println(count);  
}
```

```
}
```

Of the following, which best describes the error in `doubleUp`?

- a. The `static` modifier cannot be used with a constant.
- b. A variable is used without being initialized.
- c. A `double` cannot be multiplied by an `int`.
- d. The counter is incremented in the wrong place.
- e. A variable is initialized in the wrong place.

6. A programmer has mistakenly placed a semicolon at the end of the `while` statement in the following code segment. What will be the result of the executing code?

```
while ( j < n );    // mistaken semicolon
{
    System.out.println("in loop");
    j++;
}
```

- a. The code will behave in the same manner as if the semicolon were not included.
- b. The machine will get caught in an infinite loop, appearing to do nothing.
- c. The machine will get caught in an infinite loop, repeatedly printing the message, "in loop".
- d. Depending upon the code that precedes this loop within the program, either an infinite loop will result or the "in loop" message will be printed a single time.
- e. The code will not compile because of the presence of the semicolon.

7. Assuming that `b` is a `boolean` variable, what will be the result of executing the code?

```
b = (b != b);
```

- a. Sets `b` to false.
- b. Sets `b` to true.
- c. Changes the value of `b` to its negation.
- d. Leaves the value of `b` unchanged.
- e. The code will not compile.

8. Consider the following method:

```
public static int mystery(int n)
{
    int a, b, c = 0;
    while (n > 0)
    {
        a = n / 10;
        b = n % 10;
        c *= 10;
        c += b;
        n = a;
    }
    return c;
}
```

Which statement best describes the value that is returned by `mystery`?

- a. Ten to the n^{th} power is returned.
- b. The number of digits in `n` is returned.
- c. The number of times 10 is a factor of `n` is returned.
- d. The number whose digits are reversed of `n` is returned.

e. The sum of the digits in n is returned.

9. Consider the following class declaration:

```
public class SimpleInt
{
    private int myInt;

    public void setVal(int v)
    {
        myInt = v;
    }

    public int getVal()
    {
        return myInt;
    }
}
```

Assuming that x is a properly constructed `SimpleInt`, which of the following statements will compile in a program that uses a `SimpleInt`?

- I. `System.out.println(x.getVal());`
- II. `System.out.println(x.myInt);`
- III. `System.out.println(x.setVal(10));`

- a. I only
- b. II only
- c. I and II only
- d. I and III only
- c. I and II only

10. Which of the following cannot always be checked by a compiler?

- a. The version of a method being invoked.
- b. The spelling of keywords.
- c. Possible missing punctuation within code.
- d. Return values matching method return types.
- e. Invocation of a private method from outside the class.

11. Consider the following code segment.

```
public static int calculate(int x)
{
    x = x * x;
    x = x * x;
    return x;
}
```

What is the result of a call to compute?

- a. Returns $2y$
- b. Returns x^4
- c. Returns $2x^2$
- d. Returns $4x^2$
- f. Returns the original value of x , since the method is static

12. Which of the following is the negation of the expression `(a && !b)` ?

- a. `!a || b`
- b. `!(!a || b)`
- c. `!a && !b`
- d. `!(!a && b)`
- e. `a && b || !b`

13. Consider the method `doubleUp` that is intended to return the number of times value must be doubled before it is greater than target. `doubleUp` does not work as intended.

```
public static int doubleUp(int value, double target)
{
    int value = 1;
    int count = 0;
    while (value < target)
    {
        count++;
        value *= 2;
    }
    return count;
}
```

Of the following, which best describes the error in `doubleUp`?

- a. An extra variable is used.
- b. An uninitialized variable is used.
- c. An `int` cannot be compared to a double.
- d. A variable is incremented in the wrong place.
- e. The loop test (i.e. control expression) is incorrect.

14. Suppose that a `Widget` named `w` has been properly constructed. Which of the following expressions will compile and execute as intended without error?

- a. `System.out.println(name + " " + cost);`
- b. `System.out.println(w.name + " "+ w.cost);`
- c. `System.out.println(w.name + " " + (String) w.cost);`
- d. `System.out.println(w.getName() + " " + w.getCost());`
- e. `System.out.println(w.getName() + " " + (String) w.getCost());`

The next two questions refer to the following information.

Consider the following code segment that is intended to output a message describing a student's academic status.

```
if (grade >= 90)
    System.out.println("Excellent");
if (grade >= 80)
    System.out.println("Good");
if (grade >= 70)
    System.out.println("Passing");
if (grade >= 60)
    System.out.println("Borderline");
if (grade < 60)
    System.out.println("Failing");
```

15. If the above code is executed with a grade of 80, what will be printed?

- a. ExcellentGood
- b. Good
- c. GoodPassingBorderline
- d. GoodPassingBorderlineFailing
- e. PassingBorderline

16. The code segment is flawed. If the code is rewritten, which set of values below is best for testing its correctness?

- a. {95, 85, 75, 65, 55}
- b. {90, 89, 80, 79, 70, 69, 60, 59}
- c. {100, 90, 80, 70, 60, 50, 0}
- d. Ten values chosen randomly from 0 to 100 should be tested
- e. {99, 89, 79, 69, 59, 49} plus several random values

17. What is always true when a while loop terminates (in the absence of a break statement)?

- a. The loop control expression is true.
- b. The loop control expression is false.
- c. The variables used in the loop are undefined.
- d. The variables used in the loop are redefined as 0 or null.
- e. The body of the loop has been executed.

18. Consider the following code segment:

```
String w = <some word>;
int j = 0;
int k = w.length() - 1;
// <loop invariant>
while (j < k && w.substring(j, j + 1).equals(w.substring(k, k + 1)))
{
    j++;
    k--;
}
```

Which of the following should replace *<loop invariant>*?

- a. w[k] through w[w.length() - 1] is the reverse of w[0] through w[j]
- b. w[k] through w[w.length() - 1] is equivalent to w[0] through w[j]
- c. w[k + 1] through w[w.length() - 1] is the reverse of w[0] through w[j - 1]
- d. w[k + 1] through w[w.length() - 1] is equivalent to w[0] through w[j - 1]
- e. w[k - 1] thru w[w.length() - 1] is in alphabetical order with respect to w[0] thru w[j + 1]

19. Consider the following code segment that is intended to toggle variable x between 1 and -1. Assume x initially contains either the value 1 or -1.

```
if (x == 1)
    x = -1;
else
    x = 1;
```

- a. x += -x;
- b. x -= -x;
- c. x *= -x;
- d. x /= -x;
- e. x = -x;

20. Which assignment statement is legal?

- a. String s = "40";
- b. Integer i = 40;
- c. String s = new String("40");
- d. Integer i = new Integer(40);
- e. All of the above are legal

Part I - Free Response

1. Write a method named `remove` as started below. The method finds the first occurrence of a specified `String` named `word` in a given `String` named `text` and returns `text` with `word` removed. There should only be one space in the position where `word` is removed. In other words, there should not be any occurrences of two consecutive blank spaces. If `word` is not found, the method returns `text` unchanged.

```
// precondition: text is a non-empty string that consists of words separated by
// single blank spaces
// postcondition: if word is found in text, its first occurrence is removed from text
// and text is then returned; otherwise text is returned unchanged
public static String remove(String text, String word)
```

2. Assume that you are given the following `Sentence` class which has a class invariant that `myWords` will always contain a set of English words that are separated by single blank spaces. Each word in `myWords` only contains letters. That is, there are no punctuation symbols (such as periods, question marks, or even hyphens) or digits within `myWords`. There is no leading blank space in front of the first word in `myWords` and there is no trailing blank space behind the last word in `myWords`. You are also guaranteed as a class invariant that `myWords` will always contain at least 2 or more words.

```
public class Sentence
{
    private String myWords;

    default and "other" constructors are fully implemented and available

    // postcondition: the number of separate words in myWords is returned
    public int numWords()
    {
        a) implement this method here
    }
}
```

```
// precondition: the number of separate letters in myWords is returned
public int numLetters()
{
    b) implement this method here

}

// precondition: key will not be a null string and it will be a word
// that only contains letters
// precondition: if the String key is found anywhere within myWords,
// the boolean value true will be returned;
// otherwise false will be returned.
public boolean find(String key)
{
    c) implement this method here

}

// precondition: num > 0 and num will be less than or equal to
// the number of words in myWords
// precondition: return the numth word in myWords where the very
// first word would be returned if num is 1
public String getWord(int num)
{
    d) implement this method here

}

}
```

```
// precondition: return the boolean value true if any word
// occurs twice or more in myWords
public boolean containsDouble()
{
    e) implement this method here
```

```
}
```

```
// precondition: return the boolean value true if any word
// ends with the letter lowercase 's'
public boolean containsWordThatEndsWithS()
{
    f) implement this method here
```

```
}
```

```
}
```