Quiz#_____

AP Quiz
Java
Mr. Neat

Select the best answer. Please don't write on this quiz booklet. Good luck!

1) What is the output of the following line of code?

System.out.println(1%5 + 1%7);

a) 1
b) 2
c) 4
d) 5
e) 6

2) What is the number $(11101110)_2$ in base 10?

a) 241
b) 243
c) 245
d) 238
e) None of the above

3) Consider the following *interface* and *class*:

```
public interface Student
{
        double getGPA();
        int getSemesterUnits();
}
public class FullTimeStudent implements Student
{
        <required methods go here>
}
```

What is the minimum set of methods that a developer must *implement* in order to successfully compile the *FullTimeStudent class*?

a) none: you would not be required to implement any methods
*b) getGPA(), getSemesterUnits()*
*c) getGPA(), getSemesterUnits(), toString()*
d) *getSemesterUnits()* only
e) *getGPA()* only

4) What is the value of *v[4]* after the following code is executed?

```
int I;
int d = 1;
int[] v = {1, 1, 1, 1, 1};

for(I = 0; I < v.length; I++)
{
        d = d*2;
        v[I] = v[I] + d;
}
```

*a) 16*
*b) 32*
*c) 33*
*d) 64*
*e) 65*

5) Consider the following method where *num > 0.*

```
public static int doWhat(int num)
{
        int var = 0;
        for(int loop = 1; loop <= num; loop = loop+2)
        {
                var+= loop;
        }
        return var;
}
```

Which of the following best describes the value returned from a call to *doWhat*?

*a) num*
b) The sum of all integers between *1* and *num*, inclusive
c) The sum of all even integers between 1 and *num*, inclusive
d) The sum of all odd integers between 1 and *num*, inclusive
e) No value is returned because of an infinite loop.

6) What is the output of the following line of code?

*System.out.println("Will it ever rain\nin LA?\nHow about\nBoston?");*

a) *Will it ever rain\nin LA?\nHow about\nBoston?*
*b) Will it ever rain*
*in LA?*
*How about*
*Boston?*
c) *Will it ever rain\nin LA?*
*How about\nBoston?*
d) *Will it ever rain in LA? How about Boston?*
e) *Will it ever rain in LA? How about Boston?*

7) Consider the following code segment. Assume everything not shown compiles and runs successfully.

```
int counter = 0;
while(true)
{
      System.out.print(counter);
      if(counter > 10)
      {
                  break;
      }
}
```

What is the output?

a) 00000000000...
b) it doesn't compile
c) 12345678910
d) 0123456789
e) 012345678910

8) Consider the following code segment which compiles:

```
String quartet = new String("Happy Halloween!");
System.out.print(quartet.substring(quartet.indexOf("low")));
```

What is the output?

a) nothing
b) ween!
c) oween!
d) loween!
e) lloween!

9) What is the output of the following code segment?

```
for(int j=7;j>-1;j--)
{
        for(int k =0;k<j;k++)
        {
                System.out.print("  ");  // one space
        }
        System.out.println("*");
}
```

a) *******

b) ********

c)
```
*
 *
  *
   *
    *
     *
      *
       *
```

d)
```
********
********
********
********
********
********
********
```

e)
```
       *
      *
     *
    *
   *
  *
 *
*
```

10) Assume that x and y are boolean variables and have been properly initialized.

(x | | y) && x

Which of the following always evaluates to the same value as the expression above?

a) x
b) y
c) x && y
d) x | | y
e) x != y

11) A pair of number cubes is used in a game of chance. Each number cube has six sides, numbered from 1 to 6, inclusive, and there is an equal probability for each of the numbers to appear on the top side (indicating the cube's value) when the number cube is rolled. The following incomplete statement appears in a program that computes the sum of the values produced by rolling two number cubes.

*int sum = /* missing code */ ;*

Which of the following replacements for */ * missing code */* would best simulate the value produced as a result of rolling two number cubes?

*a) 2 * (int)(Math.random()*6)*
*b) 2 * (int)(Math.random()*7)*
*c) (int)(Math.random() * 6) + (int)(Math.random()*6)*
*d) (int)(Math.random() * 13)*
*e) 2 + (int)(Math.random() * 6) + (int)(Math.random()*6)*

12) Classes *Salsa* and *Swing implement* an *interface Dance*. If both calls

```
perform(new Salsa());
perform(new Swing());
```

are valid, which of the following could serve as the definition of the *perform* method?

I. *public void perform(Swing dance){<code not shown>}*
*II.* *public void perform(Dance dance){<code not shown>}*
*III.* *public void perform(Salsa dance){<code not shown>}*

a) I only
b) II only
c) I and II only
d) II and III only
e) I, II and III

13) Francine tests the following code below and gets no output.

```
String jj = "Next year the Red Sox will play in the World Series";
EasyReader c = new EasyReader();
String w = c.readWord();
int temp = jj.indexOf(w);
if(temp == 0)
{
        System.out.println(w+ " is @ the beginning of "+jj);
}
else if(temp == jj.length()-w.length())
{
        System.out.println(w + " is @ the end of "+jj);
}
else if(temp > 0 && temp < jj.length())
{
        System.out.println(w+" is in the middle of "+ jj);
}
```

Give an example of what she input to give no output and what should be added to the code to fix this limitation?

a) **year**, add another if-else statement
b) **play**, add an else statement at the end
c) **Red**, add another if-else statement
d) **Socks**, add an else statement at the end
e) **World**, remove the first if-else statement

14) Consider the following code segment.

```
for(int outer = 1; outer <= 6; outer++)
{
        for(int inner = outer; inner <= 6; inner++)
        {
                if(inner %2 == 0)
                {
                        System.out.print(inner + "  ");
                }
        }
        System.out.println();
}
```

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

a)
```
2       4
2       4
4
4
```

b)
```
2       4       6
2       4       6
2       4       6
2       4       6
2       4       6
2       4       6
```

c)
```
2       4       6
2       4       6
4       6
4       6
6
6
```

d)
```
2       4       6
2       4       6
2       4       6
```

e)
```
2       4       6
4       6
```

The next few questions refer to the following partial class definition:

```
public class TicketSales
{
        private String myName;
        private double[] mySales;

        public TicketSales(String movieName)
        {
                <code not shown>
        }
        public void setWeekSales(int week, double dollars)
        {
                <code not shown>
        }
        private int findBestWeek()
        {
                <code not shown>
        }
}
```

15) The reason for *private* methods like *findBestWeek* is

a) to allow clients (users) to use the method.
b) methods that work with *private* instance variables of the array type cannot be *public*.
c) to provide a method that returns an integer.
d) to provide a mechanism to repeat code within the file of the *class* definition.
e) methods that return integers cannot be *public.*

16) The constructor for the *TicketSales class* initializes the *mySales* array to hold 52 values. Which of the following statements will do that?

*a) double mySales[52];*
*b) double mySales = new double[52];*
*c) double[] mySales new double[52];*
*d) mySales = new double[52];*
*e) mySales.setSize(52);*

17) Given the declaration

*TicketSales movie = new TicketSales("Monsters, Inc.");*

Which of the following statements sets the third week sales for that movie to 245,000?

*a) movie = TicketSales(3, 245000.000);*
*b) setWeekSales(movie, 3, 245000.00);*
*c) movie.setWeekSales(3, 245000.00);*
*d) movie(setWeekSales, 3, 245000.00);*
*e) setWeekSales(3, 245000.00);*

18) Consider the following method which is intended to determine whether a letter occurs two times consecutively in a given word. The method should return true if the letter repeats in adjacent locations and false if it does not.

```
public static boolean isConsecutiveDouble(String let, String word)
{
        boolean pie = false;

        if (word.indexOf(let) >=0)
        {
                if(( word.substring(word.indexOf(let) + 1)).indexOf(let) >= 0)
                {
                        pie = true;
                }
                else
                {
                        pie = false;
                }
        }
        return pie;
}
```

Which of the following is true?

a) the method would work incorrectly for the call *isConsecutiveDouble("t", "butter");*
b) the method would work incorrectly for the call *isConsecutiveDouble("t", "tutor");*
c) the method would work incorrectly for the call isConsecutiveDoulbe("t", "ttttt");
d) the method would work incorrectly for the call isConsecutiveDouble("t", "but");
e) the method works as intended for all cases.

19) Consider the following complete Java program.

```
class temp
{
        public static void main(String args[])
        {
                int[] temp = mystery(6);
                System.out.println(temp[5]);
        }
        public static int[] mystery(int n)
        {
                int[] seq = new int[n];
                for(int k = 0; k < n ; k++)
                {
                        seq[k]= k*k+3;
                }
                return seq;
        }
}
```

Which of the following is the output of the program?

a) 28
b) 19
c) 12
d) Exception in thread “main” java.lang.ArrayIndexOutOfBoundsException: 6
at temp.main(temp.java:8)
e) none of the above

20) Consider the following method.

```
public static int mystery(int[] arr)
{
        int index =0;
        int count = 0;
        int m = -1;

        for(int outer=0; outer< arr.length; outer++)
        {
                count = 0;
                for(int inner = outer+1; inner < arr.length; inner++)
                {
                        if(arr[outer] == arr[inner])
                        {
                                count++;
                        }
                }
                if(count > m)
                {
                        index = outer;
                        m = count;
                }
        }
        return index;
}
```

Assume the nums has been declared and initialized as an array of integer values. Which of the following best describes the value returnded by the call mystery(nums)?

a) The maximum value that occurs in nums
b) An index of the maximum value that occurs in nums
c) The number of times that the maximum value occurs in nums
d) A value that occurs most often in nums
e) An index of a value that occurs most often in nums.

21) What is the output of the following program? What would be a descriptive name for the method *mystery*?

```
public static void main(String args[])

{
        System.out.println(mystery(17,3));
}
public static int mystery(int a, int b)
{
        int diff = a - b;
        int counter = 0;

        if(a==b)
        {
                return 1;
        }
        else if(a < b)
        {
                return 0;
        }
        else
        {
                while(diff >= 0)
                {
                        counter++;
                        a = a - b;
                        diff = a - b;
                }
        }
        return counter;
}
```

a) 2, divideInt
b) 5, moduloInt
c) 5, divideInt
d) 2, moduloInt
e) 14, subtractInt

22) Consider the following global variable and method.

```
private int[] numbers;

public void mystery(int x)
{
        for(int k=1;k<numbers.length;k=k+x)
        {
                numbers[k] = numbers[k-1] +x;
        }
}
```

Assume that *numbers* has been initialized with the following values.

{17, 34, 21, 42, 15, 69, 48, 25, 39}

Which of the following represents the order of the values in *numbers* as a result of the call *mystery(3)*?

a) {17, 20, 21, 42, 45, 69, 48, 51, 39}
b) {17, 20, 23, 26, 29, 32, 35, 38, 41}
c) {17, 37, 21, 42, 18, 69, 48, 28, 39}
d) {20, 23, 21, 42, 45, 69, 51, 54, 39}
e) {20, 34, 21, 45, 15, 69, 51, 25, 39}

The next set of questions refer to the Worker class below:

```
public class Worker
{
        private String name;
        private double hourlyWage;
        private boolean isUnionMember;

        public Worker()
        {
                // implementation not shown
        }
        public Worker(String aName, double anHourlyWage, boolean union)
        {
                // implementation not shown
        }

        public String getName()
        {
                // getName missing code
        }
        public double getHourlyWage()
        {
                // getHourlyWage missing code
        }
        public void incrementWage(double amt)
        {
                // incrementWage  missing code
        }
        public void changeUnionStatus()
        {
                // changeUnionStatus missing code
        }
}
```

23) How many methods is(are) there in the Worker class?

a) 0
b) 1
c) 2
d) 3
e) 4

24) Which of the following is a correct replacement for incrementWage missing code?

a) *return hourlyWage + amt;*
b) *return getHourlyWage() + amt;*
c) *hourlyWage += amt;*
d) *getHourlyWage() += amt;*
e) *hourlyWage = amt;*

25) How many constructors is(are) there in the Worker class?

a) 0
b) 1
c) 2
d) 3
e) 4

26) Which of the following correctly constructs an instance of the Worker class?

a) *Worker person = new Worker(Sam, 11.55, false);*
b) *Worker person = new Worker("Sam", 11.55);*
c) *Worker person = new Worker();*
d) *Worker person = new Worker("Sam");*
e) *Worker person = new Worker("Sam",true,11.55);*

27) Consider the method changeUnionStatus. Which is a correct replacement for *changeUnionStatus missing code?*

I. *isUnionMember = !isUnionMember;*

II.
```
if(isUnionMember)
{
        isUnionMember = !isUnionMember;
}
```

III.
```
if (isUnionMember)
{
        isUnionMember = false;
}
else
{
        isUnionMember = true;
}
```

a) I and II only
b) I and III only
c) II and III only
d) I, II and III
e) I only

28) A client method (in another file) computePay will return a worker's pay based on the number of hours worked.

```
public static double computePay(Worker w, double hours)
{
        // computePay missing code
}
```

Which replacement for *computePay* missing code is correct?

a) *return w.getHourlyWage()*hours;*
b) *return w.hourlyWage*hours;*
c) *return w.getHourlyWage()*w.hours;*
d) *return hourlyWage;*
e) *return getHourlyWage()*hours;*

29) The following method is intended to return the index of the “smallest” String, namely the String that would appear first in a sorted list of String objects.

```
public static int findMin(String[] books)
{
        int minPos = 0;
        for(int index = 1; index < books.length; index++)
        {
                if( /* condition */)
                {
                        minPos = index;
                }
        }
        return minPos;
}
```

Which of the following should be used to replace /* condition */ so that findMin works as intended?

a) *books[index].compareTo(books[minPos]) < 0*
b) *books[index].compareTo(books[minPos]) > 0*
c) *books[minPos].compareTo(books[index]) < 0*
d) *books[index] < books[minPos]*
e) *books[index] > books[minPos]*

30) What does the following code segment do?

```
public static boolean mystery(String s)
{
        String a = "";
        for(int i = s.length()-1; i >=0; i--)
        {
                a+=s.substring(i,i+1);
        }
        return s.equals(a);
}
```

a) the method *mystery* determines whether the *String s* has a double letter.
b) the method *mystery* determines whether the *String s* is reversed.
c) the method *mystery* returns the reversed *String s*
d) the method *mystery* determines whether the *String s* has an adjacent double letter.
e) none of the above

Short Answer: Provide descriptive (one word) names for each of the methods in the following complete Java program. What is the output? Write your answers on the provided answer sheet.

```
public class starter {

	public static void main(String[] args)
	{
		int[] zero = {7, 3, 5, 8, 9};
		four(zero);
		zero = one(zero);
		four(zero);
		System.out.println(tree(zero));

	}
	public static int[] one(int[] es)
	{
		int what=0;
		for(int i=0; i< es.length;i++)
		{
			es[i]=two(i,es[i]);
		}
		return es;
	}

	public static int two(int a, int b)
	{
		int[] trash = new int[a];
		for(int j=0;j<trash.length;j++)
		{
			trash[j] = b;
		}
		return a*b;
	}
	public static int tree(int[] tee)
	{
		int whatzup = 0;
		for(int k=0;k<tee.length;k++)
		{
			whatzup = whatzup + tee[k];
		}
		return whatzup;
	}
	public static void four(int[] hee)
	{
		System.out.println();
		for(int m=0;m<hee.length;m++)
		{
			System.out.print(hee[m] + " ");
		}
		System.out.println();
	}
}
```