Level 4 Practice Exam

Name: _____

1. What is the output of the following code?

```
int[][] arr = new int[5][5];
for(int i = 0; i < arr.length; i++){
      for(int j = 0; j < arr[i].length; j++){
            arr[i][j] = i * 5 + j;
      }
}

System.out.println(arr[1][4]);
```

9

2. Rewrite the following if/else block with the same logic using a switch-statement.

```
int x = new Random().nextInt(3);
if(x == 0){
      System.out.println("cheese");
}else if(x == 1){
      System.out.println("pepperoni");
}else if(x == 2){
      System.out.println("sausage");
}else{
      System.out.println("ERROR!");
}
```

```
int x = new Random().nextInt(3);
switch(x):
  case(0) {
    System.out.println(“cheese”);
    break;
  }
  case(1) {
    System.out.println(“pepperoni”);
    break;
  }
  case(2) {
    System.out.println(“sausage”);
    break;
  }
  default {
    System.out.println(“ERROR!”);
```

3. Create two objects of the CustomList class. One that defines T as a String and one that defines it as a Random.

```
class CustomList<T>{

}
```

4. Use the Season enum and the s variable to complete the if/else statements in the printSeasonalGreeting method.

```
enum Season{
          WINTER, SPRING, SUMMER, FALL
}


void printSeasonalGreeting(Season s){
     if      (                                        ){
          System.out.println("Happy Snowy Holidays!");
     }else if(                                        ){
          System.out.println("Flowers are blooming!");
     }else if(                                        ){
          System.out.println("IT'S TOO HOT!");
     }else if(                                        ){
          System.out.println("Leaves are falling off the trees.");
     }
}
```

5. Write a method called reverseString that takes a String as a parameter and returns the backwards version of that string.

Example:

```
reverseString("Nachos"); //should return a String equal to "sohcaN"
```

6. What is the output of the following code if we assume the user entered input that is NOT a number?

```
int x = 0;
String in = JOptionPane.showInputDialog("Enter a number");
try{
      System.out.println("trying…");
      x = Integer.parseInt(in);
      System.out.println("success!");
      x = 1;
}catch(Exception e){
      System.out.println("failure!");
}

System.out.println("x = " + x);
```

7. Encapsulate the value variable in the Account class below. Make sure it can never be set to a negative number.

```
public class Account{
      int value;
      public Account(int v){
            value = v;
      }









}
```

8. Explain the error in the following code?

```
class Super{}
class Sub extends Super{}
public class Runner(){
     public static void main(String[] args){
          Super s = new Super();
          Super t = new Sub();
          Sub u = new Sub();
          Sub v = new Super();
     }
}
```

9. Fill in the code so that it prints the elements at the 3rd row and the 2nd column.

```
String[][] words = new String[10][10];
// words gets populated with strings here

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

10. What would be the output produced by the following code?

```
System.out.println("A comment looks like this:\n\\*comment here*\\");
```

11. Add code to the following so that the String out will be the same value as in, except with the characters alternating between capital and lowercase.

```
String in = JOptionPane.showInputDialog("Enter some words.");
String out = "";
// code goes here
```

12. Explain why the following code will not compile.

```
try{
      Thread.sleep(100);
}
```

13. What is the value of the array nums after the following code executes?

```
int[] nums = { 9, 2, 4, 1, 3, 5 };
for(int i = 0; i < nums.length / 2; i++){
      int t = nums[i];
      nums[i] = nums[nums.length - 1 - i];
      nums[nums.length - 1 - i] = t;
}
```

nums is equal to: [ ][ ][ ][ ][ ][ ]

14. A window is displayed at 500 x 500 pixels. The window is evenly divided by into a 10 x 10 grid. The mouse is clicked at pixel x: 325 y: 212. If the top left grid square is located at x: 0 y:0, what is the location of the grid square that was clicked?

x: _____

y: _____

15. Add code to the Circle class so that it will compile without errors.

```
public interface Shape
{
    int isLargerThan(Shape other);
}
public class Circle implements Shape
{
    // Add code here
```

```
}
```

16. Complete the oddArray method below so that it turns every odd integer in arr to 0.

Example:

```
     int[] a = {2, 3, 5, 6, 8, 9};
     oddArray(a); // a is now equal to: {2, 0, 0, 6, 8, 0};

public void oddArray (int[] arr)
{









}
```

17. What is returned when the numFun method is called with the value of 21560?

```
public static int numFun(int num)
{
     if (num / 10 == 0){
               return num;
     } else {
               return (num % 10) + numFun(num / 10);
     }
}
```

18. What are the contents of mat after the following code segment has been executed?

```
int[][] mat = new int[3][4];
for (int row = 0; row < mat.length; row++)
{
    for (int col = 0; col < mat[i].length; col++)
    {
        if (row < col)
        {
            mat[row][col] = 1;
        }
```

```
        else if (row == col)
      {
            mat[row][col] = 2;
        }
        else
        {
            mat[row][col] = 3;
        }
    }
}
```

mat is equal to [ ][ ][ ][ ]
[ ][ ][ ][ ]
[ ][ ][ ][ ]

19. What is printed after the following code executes?

```
String codeAllDay = "Code all day.";
int indexOfFirstSpace = codeAllDay.indexOf(" ");
int indexOfSecondSpace = codeAllDay.indexOf(" ", indexOfFirstSpace + 1);
codeAllDay.substring(0, indexOfSecondSpace);
String codeAllNight = codeAllDay + " night";
System.out.println(codeAllDay + codeAllNight);
```

20. Consider the following two classes.

```
public class Dog {
      public void act() {
            System.out.print("run");
            eat();
      }

      public void eat() {
            System.out.print("eat");
      }
}
```

```
public class UnderDog extends Dog {
      public void act() {
            super.act();
            System.out.print("poop");
      }

      public void eat() {
            super.eat();
            System.out.print("bark");
      }
}
```

Assume that the following declaration appears in a client program.

```
      Dog fido = new UnderDog();
```

What is printed as a result of the call `fido.act();` ?