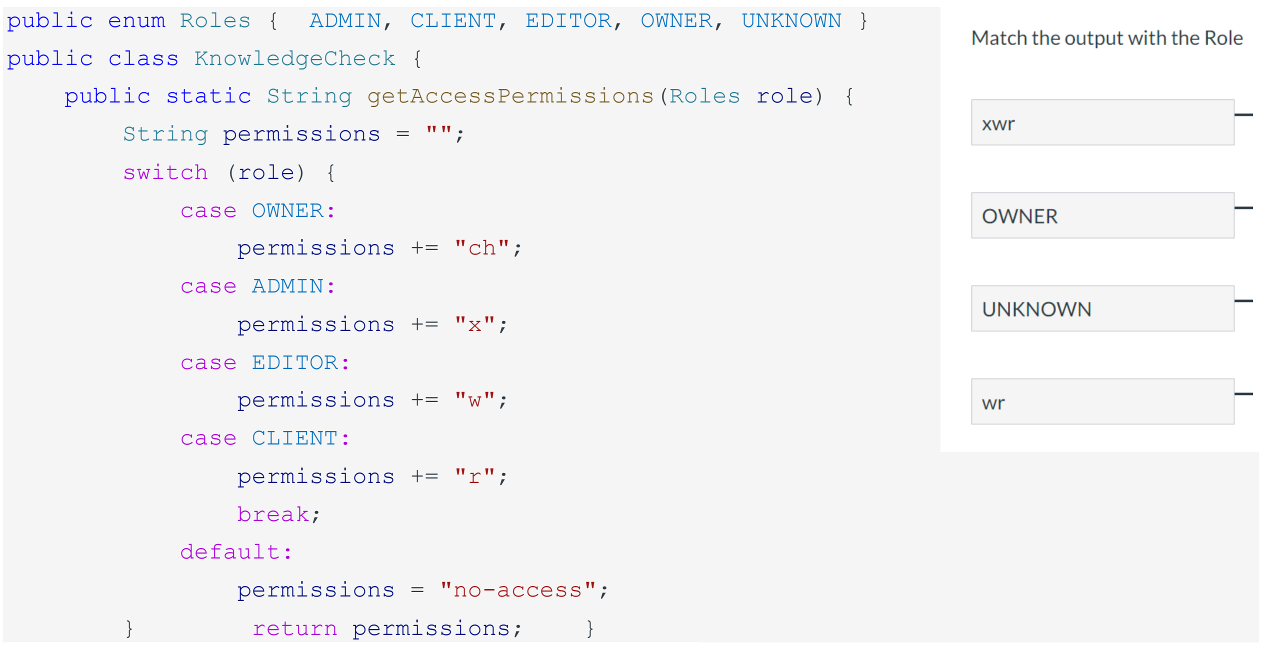
C163/164: Exam 3 Review Worksheet 2
Name(s):____

1. Write exactly what this program is going to print.

```
public interface Shape {
    int getWidth();
    int getLength();
    int getArea();
}
public abstract class AbstractShape implements Shape {
    private int width;
    private int length;
    public int getWidth() {
        return width;
    }
    public int getLength() {
        return length;
    }
    public void setWidth(int val) {
        width = val;
    }
    public void setLength(int val) {
        length = val;
    }
    public AbstractShape(int width, int length) {
        setWidth(width);
        setLength(length);
    }
}
public class Rectangle extends  AbstractShape{
    public Rectangle(int width, int length) {
        super(width, length);
    }
    public int getArea() {
        return getLength()*getWidth();
    }
    public String toString() {
        return String.format("Rectangle: width: %d, length: %d, area: %d", getWidth(), getLeng
th(), getArea());
    }
}
public class Triangle extends AbstractShape {
    private double angle;
    public int getArea() {
        return (getWidth()*getLength())/2;
    }
    public double getAngle() {
        return angle;
    }
    public void setAngle(double val) {
        angle = val;
    }
    public Triangle(int base, int height) {
        this(base, height, 90.0);
    }
    public Triangle(int base, int height, double angle) {
        super(base, height);
        setAngle(angle);
    }
    public String toString() {
        return String.format("Triangle: base: %d, height: %d, angle: %.2f, area: %d",
                getWidth(), getLength(), getAngle(), getArea());
    }
}
```

C163/164: Exam 3 Review Worksheet 2
Name(s):____

```
public class Square extends Rectangle {
   public Square(int side) {
        this(side, side);
    }
   private Square(int width, int length) {
        super(width, length);
    }
}
public class Main24 {
    public static void main(String[] args) {
        List<Object> shapes = new ArrayList<>();
        shapes.add(new Square(10));
        shapes.add(new Triangle(10, 5));
        shapes.add(new Rectangle(11,3));
        shapes.add(new Rectangle(10,2));
        shapes.add(new Triangle(10, 5, 23.33333));
        for(Object obj : shapes) System.out.println(obj);
 }
}
```



```
public enum Roles {  ADMIN, CLIENT, EDITOR, OWNER, UNKNOWN }
public class KnowledgeCheck {
    public static String getAccessPermissions(Roles role) {
        String permissions = "";
        switch (role) {
            case OWNER:
                permissions += "ch";
            case ADMIN:
                permissions += "x";
            case EDITOR:
                permissions += "w";
            case CLIENT:
                permissions += "r";
                break;
            default:
                permissions = "no-access";
        }          return permissions;     }
```

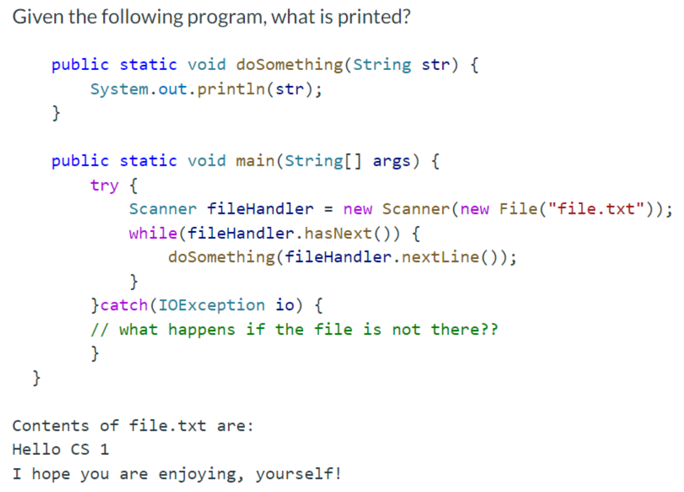
Match the output with the Role

xwr —

OWNER —

UNKNOWN —

wr —

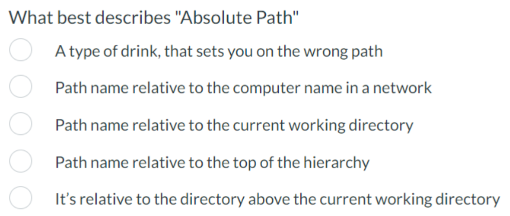


Given the following program, what is printed?

```
    public static void doSomething(String str) {
        System.out.println(str);
    }

    public static void main(String[] args) {
        try {
            Scanner fileHandler = new Scanner(new File("file.txt"));
            while(fileHandler.hasNext()) {
                doSomething(fileHandler.nextLine());
            }
        }catch(IOException io) {
        // what happens if the file is not there??
        }
  }

Contents of file.txt are:
Hello CS 1
I hope you are enjoying, yourself!
```



What best describes "Absolute Path"

- ○ A type of drink, that sets you on the wrong path
- ○ Path name relative to the computer name in a network
- ○ Path name relative to the current working directory
- ○ Path name relative to the top of the hierarchy
- ○ It's relative to the directory above the current working directory