Worksheet 3

1. C

2. C

3. C

4. C

5. D

6. C

7. C

8. D

9. C

It will output "This" because args[0] corresponds to the first word after "Output" in the command line. Therefore, the correct answer is C. This.

10. B

Adding 2.5 to this random number results in a value between 2.5 and 3.5. The Math.round() method then rounds this value to the nearest integer. Therefore, the value of d will be 3 (option B).

11. D

12. D

13. B

the variable x is declared twice: once in the outer scope and once inside the if block. However, Java uses block-level scope, so the inner declaration of x shadows the outer one only within the if block. Therefore, inside the if block, x is 8 and will be printed. Outside the if block, x remains 9

14. B

15. D

16. B

int x, y = 1; where y is initialized to 1.

x is assigned the value 10.

The if condition x != 10 && x / 0 == 0 evaluates to false because x is 10 and x / 0 causes a division by zero error, which is caught at runtime.

Therefore, the else block executes, incrementing y to 2, which is then printed.

17. D

Since volume is not properly declared or initialized within the area class, attempting to access obj.volume will likely result in a compilation error (option D) due to a variable not being initialized.

18. <access_modifier> <static_or_non_static> <return_type> <method_name>(<parameter_list>) {

```
   // Method body (statements)
}
```

19. A.

```
class Addition {
   int sum = 0;
   int addTwoInt(int a, int b) {
      sum = a + b;
      return sum;
   }
   public static void main(String[] args) {
      Addition obj = new Addition();
      int result = obj.addTwoInt(5, 3);
      System.out.println("Sum: " + result);
   }
}
```

19. B

```
class Addition {
   public int add(int a, int b) {
      return a + b;
   }
}
public class MethodCall {
   public static void main(String[] args) {
      Addition obj = new Addition();
```

```
        int result = obj.add(5, 3)
        System.out.println("Sum: " + result);
    }
}
```

20. A

```
class Example {
    private int number;
    private String name;
    public Example(int number, String name) {
        this.number = number;
        this.name = name;
    }
    public int getNumber() {
        return number;
    }
    public String getName() {
        return name;
    }
    public void setNumber(int number) {
        this.number = number;
    }
    public void setName(String name) {
        this.name = name;
    }
    public void printDetails() {
        System.out.println("Name: " + name);
        System.out.println("Number: " + number);
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        Example obj = new Example(10, "John Doe");

        System.out.println("Name retrieved using getter: " + obj.getName());
        System.out.println("Number retrieved using getter: " + obj.getNumber());

        obj.setName("Jane Smith");
        obj.setNumber(20);

        obj.printDetails();
    }
}
```

Output - Sum: 8

20. B

```
class Example {
    private int number;
    private String name;
        public Example(int number, String name) {
        this.number = number;
        this.name = name;
    }

    public int getNumber() {
        return number;
    }

    public String getName() {
        return name;
```

```
    }

    public void setNumber(int number) {
        this.number = number;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void printDetails() {
        System.out.println("Name: " + name);
        System.out.println("Number: " + number);
    }
}

public class Main {
    public static void main(String[] args) {
        Example obj = new Example(10, "John Doe");

        System.out.println("Name retrieved using getter: " + obj.getName());
        System.out.println("Number retrieved using getter: " + obj.getNumber());

        obj.setName("Jane Smith");
        obj.setNumber(20);

        obj.printDetails();
    }
}
```

output -

Name retrieved using getter: John Doe

Number retrieved using getter: 10

Name: Jane Smith

Number: 20