

**Question - 1**  
Question 1

SCORE: 20 points

What's the output of this segment?

```
1. public class A {  
2.     int add(int i, int j){  
3.         return i+j;  
4.     }  
5. }  
6. public class B extends A{  
7.     public static void main(String argv[]){  
8.         short s = 9;  
9.         System.out.println(add(s,6));  
10.    }  
11.}
```

☐ Compile fail due to error on line no 2

Compile fail due to error on line no 9 due to the non-static method referenced from a static context.

☐ Compile fail due to error on line no 9 due to type mismatch.☐ 15**Question - 2**  
Question 2

SCORE: 20 points

```
public class Q2 {  
    public static void main(String []args) {  
        ArrayList<String> list = new ArrayList<>  
        ();  
        list.add("2");  
        list.add("0");  
        list.add("1");  
        list.add("8");  
        List<String> newList = next(list);  
        list.add("9");  
        for (String i: newList) {  
            System.out.print(i + " ");  
        }  
        private static List<String>  
        next(List<String>list) {
```

```

        list.remove(list.size()-1);
        return list;
    }
}

```

- ☐ "2" "0" "8" "9"
- ☐ 2 0 1
- ☐ 2 0 8 9
- ☒ 2 0 1 9

### Question - 3

#### Question 3

SCORE: 20 points

What's the output of this segment?

```

class MyStatic {
    static int x = 6;
    MyStatic() {
        x++;
    }
    void method() {
        System.out.print("-x" + x);
    }
    public static void main(String[] args) {
        MyStatic mc1, mc2, mc3, mc4;
        MyStatic mc5 = new MyStatic();
        MyStatic mc6 = new MyStatic();
        MyStatic mc7 = new MyStatic();
        mc7.method();
    }
}

```

- ☐ -x6
- ☐ -x13
- ☒ -x9
- ☐ -x7
- ☐ Running time Error
- ☐ throw NullPointerException

### Question - 4

#### Question 4

SCORE: 20 points

What's the output of this segment?

```

class Balloon{
    private String color;
    public Balloon(){}
}

```

```

public Balloon(String c){
    this.color = c;
}
public String getColor(){
    return this.color;
}
public void setColor(String color) {
    this.color = color;
}
private static void foo(Balloon balloon){
    balloon.setColor("Red");
    balloon = new Balloon("Green");
    balloon.setColor("Blue");
}
public static void swap(Balloon o1,Balloon
o2){
    Balloon temp = o1;
    o1 = o2;
    o2 =temp;
}

public static void main(String[] args) {
    Balloon red = new Balloon("Red");
    Balloon blue = new Balloon("Blue");
    swap(red,blue);
    System.out.println("red
color="+red.getColor());
    System.out.println("blue
color="+blue.getColor());
    foo(blue);
    System.out.println("blue
color="+blue.getColor());
}
}

```

- ☐ red color=Red blue color=Blue blue color=Blue
- ☐ red color=Blue blue color=Red blue color=Red
- ☒ red color=Red blue color=Blue blue color=Red
- ☐ red color=Blue blue color=Blue blue color=Blue

### Question - 5

SCORE: 20 points

Java: Constructor, Exception Handling and Inheritance

Language Proficiency

Constructors

Java

Exception Handling

Inheritance

Core CS

What should be the output of the following code?

```

class AirPlane {
    public AirPlane() throws IOException {
        System.out.print("AirPlane");
        throw new IOException();
    }
}
class AirJet extends AirPlane {
    public AirJet() throws IOException {
        System.out.println("Airjet");
        try {
            super();
        }
    }
}

```

```

        } catch (IOException e) {
            System.out.print("IOException is
thrown in AirJet");
        }
    }
}
public class Tester {
    public static void main(String args[]) {
        try {
            new AirJet();
        } catch (IOException e) {
            System.out.print("IOException is
thrown in Tester");
        }
    }
}

```

☐ "AirPlaneIOException is thrown in AirJet" will be printed.



"AirPlaneIOException is thrown in AirJetIOException is thrown in Tester" will be printed.



"AirPlaneIOException is thrown in Tester" will be printed.



The code does not compile because of an error in the constructor of class Airjet.

### Question - 6

SCORE: 20 points

#### Question 6

Say that there are three classes: Computer, AppleComputer, and IBMComputer. What are the likely relationships between these classes?



Computer is the superclass, AppleComputer and IBMComputer are subclasses of Computer.



Computer, AppleComputer and IBMComputer are sibling classes.



IBMComputer is the superclass, AppleComputer and Computer are subclasses of IBMComputer.



Computer is a superclass, AppleComputer is a subclasses of Computer, and IBMComputer is a subclass of AppleComputer.

### Question - 7

SCORE: 50 points

#### Car Inheritance

Inheritance

Polymorphism

Easy

Abstraction

Java

C++

C#

Object Oriented Programming

Java 8

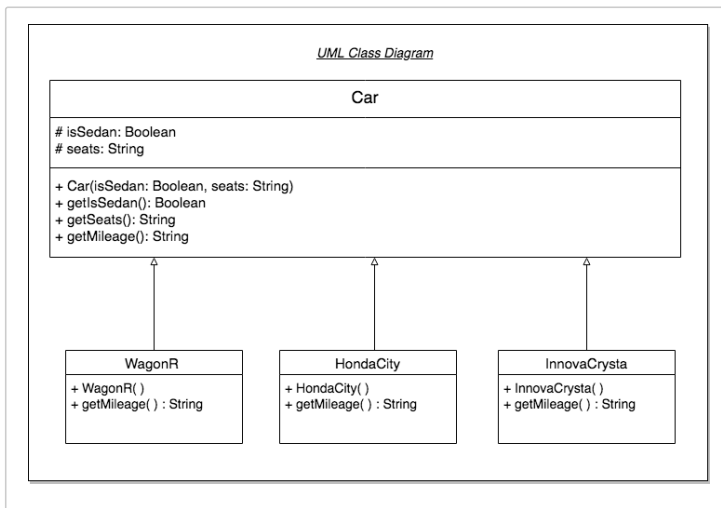
Problem Solving

Core Skills

In this challenge, you will be asked to build on an abstract class and initialize an instance of each class with a variable. The program will then test your implementation by retrieving the data you stored.

The locked code in the editor does the following:

1. Declares an abstract class named `Car` with the implementations for `getIsSedan()` and `getSeats()` methods, as well as an abstract method named `getMileage()`.
2. Creates `WagonR`, `HondaCity`, or `InnovaCrysta` object based on input (0 for `WagonR`, 1 for `HondaCity` and 2 for `InnovaCrysta`).
3. Calls the `getIsSedan()`, `getSeats()`, and `getMileage()` methods on the object.



The details for each car are provided below -

1. `WagonR` is not a sedan and has 4 seats.
2. `HondaCity` is a sedan and has 4 seats.
3. `InnovaCrysta` is not a sedan and has 6 seats.

Complete the code in the editor below to implement the following:

1. Three classes named, `WagonR`, `HondaCity`, and `InnovaCrysta` that inherit from the `Car` class.
2. One integer argument is provided to the constructor which is the mileage of the car.
3. Each class must implement the `getMileage()` method which returns a string in the form of `<mileage> kmpl` where `<mileage>` is the value provided to constructor.

#### Constraints

- The integer in first line will be in between 0 and 2 inclusive.
- The integer in the second line (i.e. mileage) will be in between 5 and 30 inclusive.

#### ▼ Input Format For Custom Testing

The first line contains an integer describing the type of car to instantiate.

The second line contains an integer, the mileage of the car.

#### ▼ Sample Case 0

##### Sample Input For Custom Testing

```
0
22
```

**Sample Output**

```
A WagonR is not Sedan, is 4-seater, and has a
mileage of around 22 kmpl.
```

**▼ Sample Case 1****Sample Input For Custom Testing**

```
1
12
```

**Sample Output**

```
A HondaCity is Sedan, is 4-seater, and has a
mileage of around 12 kmpl.
```