Name : Hardik
Std Id : 202103032

1. Identify the incorrect line from the code given and justify your answer.

```
public class StaticTest {
    StaticTest sd = new StaticTest();
    void method1() {
        method4(); // 1
        StaticTest.method2(); // 2
        StaticTest.method3(); // 3
    }
    static void method2() {
    }
    void method3() {
        method1(); //4
        method2(); //5
        sd.method2(); //6
    }
    static void method4() {
    }
}
```

Ans : In following code line 2 and 3 are incorrect because we are not allowed to use class name for call the methodes.

2. Will the given code compile successfully? Justify Yes or No.

```
  class FinalCheck {
  static final int V1 = 4;
    public static void main(String args[])
    {
        V1 = 5;
    }
}
```

Ans : No, given code will give compiler error because we can not assign value to final variable V1

3. Demonstrate the use of ‘final’ keyword to prevent the method overriding taking an example of inheritance.

```
class Vehicle {
   final void vehicle() {
        System.out.println("Vehicle");
    }
}
class Bike extends Vehicle {
    void vehicle(){
        //error : can not over-ride because in parent class
method vehicle() is final
        System.out.println("Car");
    }
}
public class StaticTest {
    public static void main(String[] args)
    {
        // Create an object Bike
        Bike bike = new Bike();
    }
}
```

Ans : Above code will give an error because we can not over-ride final method,
where I have mentioned comment it will give an error

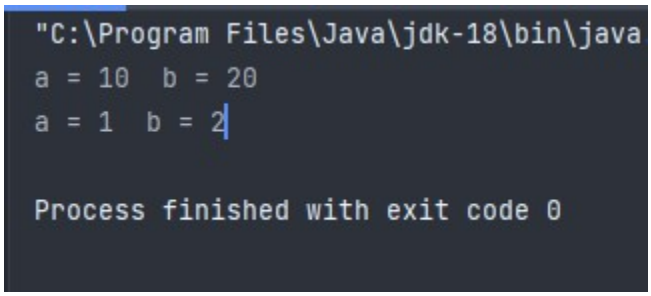
4. Show the use of ‘this’ reference variable to refer to the current class instance variable using some parameterized constructor.

Code :

```
class Test
{
    int a;
    int b;
    // Parameter constructor
    Test(int a, int b)
    {
        this.a = a;
        this.b = b;
    }
    void display()
    {
        System.out.println("a = " +this.a + "  b = " + this.b);
```

```
    }
    public static void main(String[] args)
    {
        Test object1 = new Test(10, 20);
        Test object2 = new Test(1, 2);
        object1.display();
        object2.display();
    }
}
```

Output :



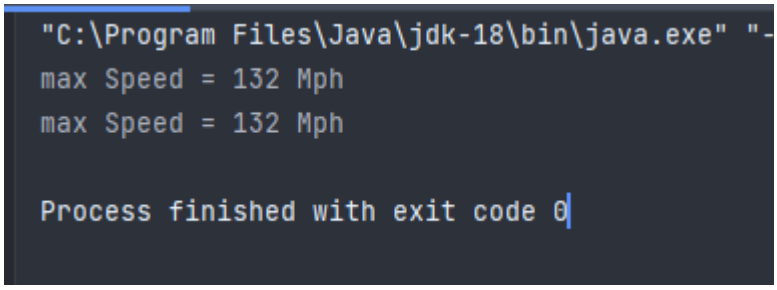
5. How can we resolve the ambiguity (methods with the same name) in the inheritance with the help of 'super' keyword? Demonstrate with an example.

Code:

```
class Car
{
    int maxSpeed = 132;
    void display(){
        System.out.println("max Speed = " + maxSpeed + " Mph");
    }
}
class BMW extends Car{
    int maxSpeed = 300;
    void display(){
        System.out.println("max Speed = " + super.maxSpeed + "
Mph");
    }
}
public class StaticTest{
    public static void main(String[] args) {
        Car car = new Car();
        Car car1 = new BMW();
        car.display();
        car1.display();
    }
```

```
}
```

Output :



6. Identify the error in the given code:

```
  class Vehicle {
 Vehicle() { System.out.println("Vehicle."); }
}
class Bike extends Vehicle {
    Bike() { System.out.println("Bike."); }
    Bike(String brand)
    {
        super();
        this();
        System.out.println("Bike is " + brand);
    }
}
public class GFG {
    public static void main(String args[])
    {
        Bike bike = new Bike("Apache");
    }
}
```

Ans : In following code we have used super() and this() in the same constructor,
If we want to use super() and this() than we have to write both of them on the first position inside the constructor which is not possible.