Inheritance

We need to explore examples on types of inheritance.

- **Single inheritance**

Example #1

```
package com.javabykiran;

public class A {
	public void methodA() {
		System.out.println("Base class method");
	}
}
```

javabyKiran
java | selenium | python

```
package com.javabykiran;

public class B extends A {
	public void methodB() {
		System.out.println("Child class method");
	}

	public static void main(String args[]) {
		B obj = new B();
		obj.methodA(); // calling super class method
		obj.methodB(); // calling local method
	}
}
```

Example #2

```
package com.jbk;

public class Vehicle {
	String vehicleType;
}
```

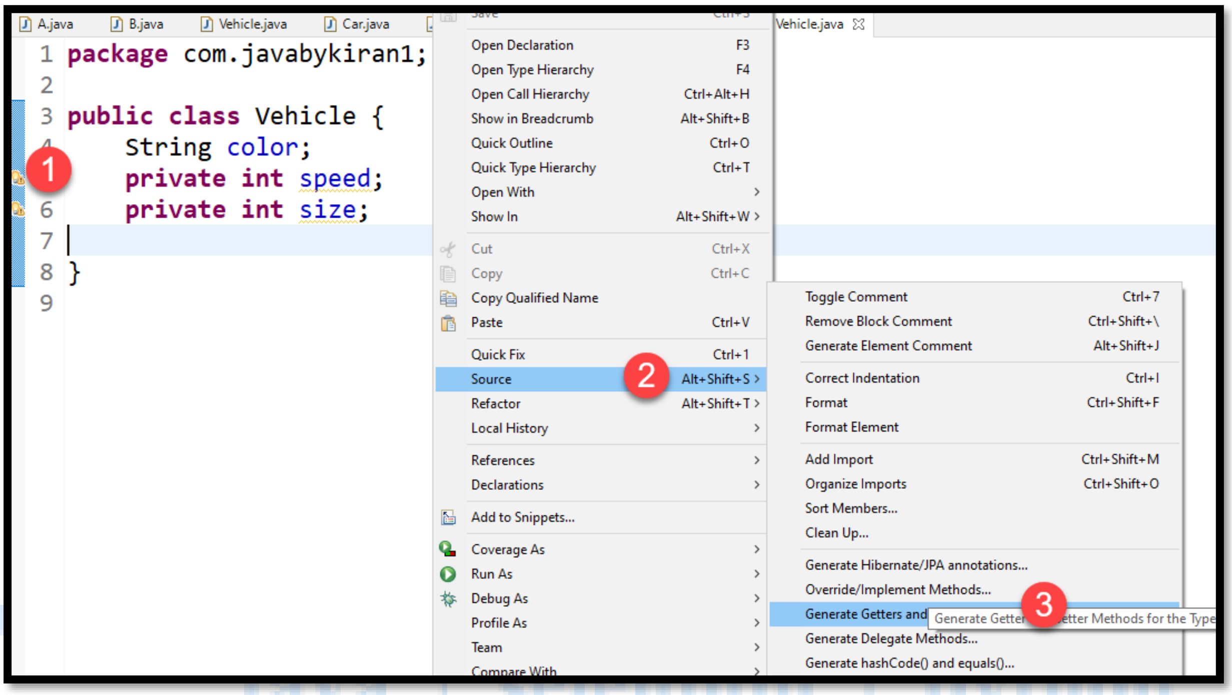
```
package com.jbk;

public class Car extends Vehicle {
	String modelType;

	public void showDetail() {
		vehicleType = "Car"; // accessing Vehicle class member
		modelType = "sports";
		System.out.println(modelType + " " + vehicleType);
	}

	public static void main(String[] args) {
		Car car = new Car();
		car.showDetail();
	}
}
```

javabyKiran java | selenium | python

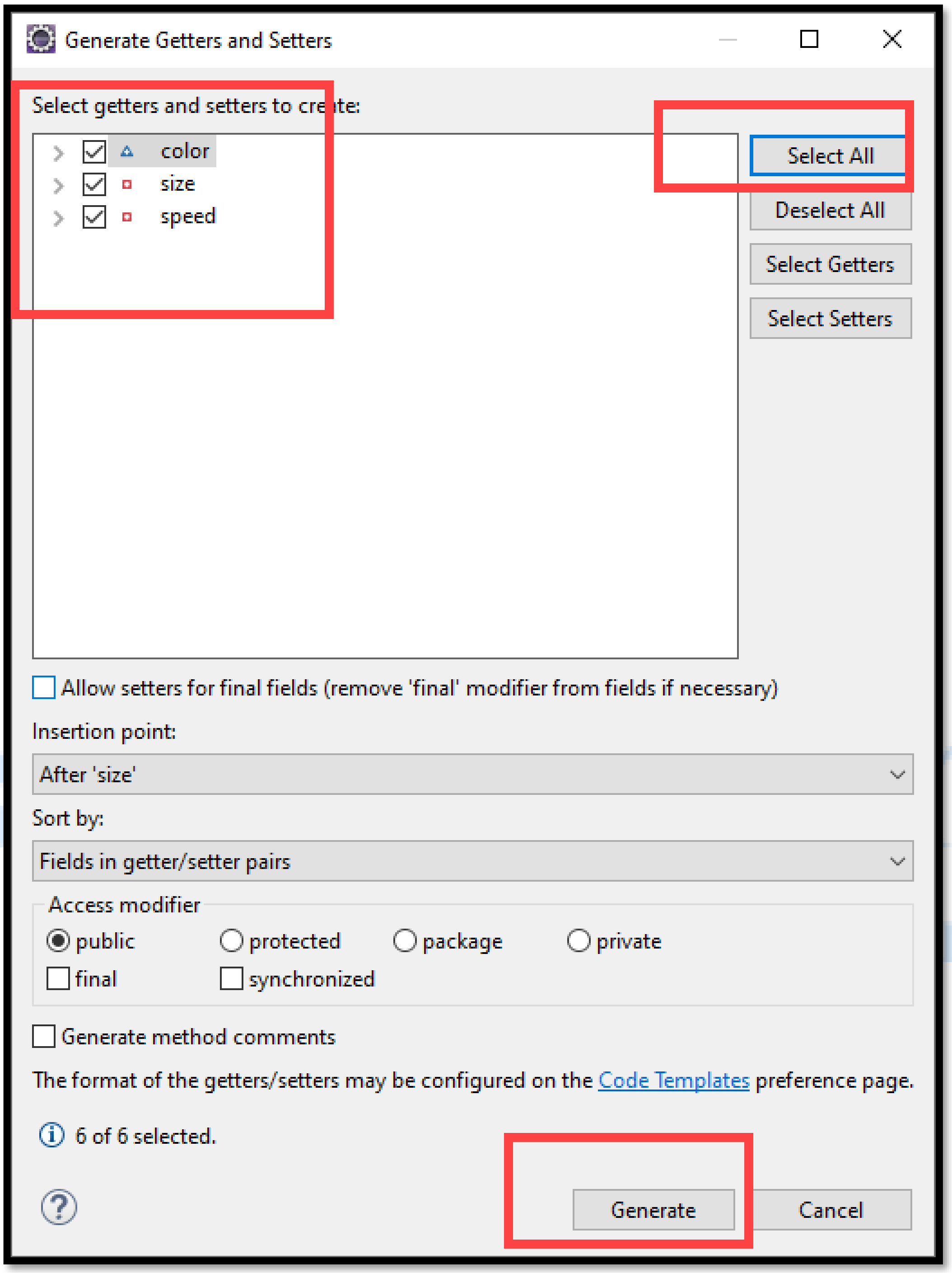
Example #3

Program with getters and setters

We can generate that automatically by following below steps. first declare variables then right click



```
package com.javabykiran1;

public class Vehicle {
    String color;
    private int speed;
    private int size;

}
```


Generate Getters and Setters

Select getters and setters to create:

- [✓] color
- [✓] size
- [✓] speed

[ ] Allow setters for final fields (remove 'final' modifier from fields if necessary)

Insertion point: After 'size'

Sort by: Fields in getter/setter pairs

Access modifier: [✓ public] protected package private

[ ] final [ ] synchronized

[ ] Generate method comments

The format of the getters/setters may be configured on the Code Templates preference page.

6 of 6 selected.

```
package com.javabykiran1;

public class Vehicle {
	String color;
	private int speed;
	private int size;

	public String getColor() {
		return color;
	}

	public void setColor(String color) {
		this.color = color;
	}

	public int getSpeed() {
		return speed;
	}

	public void setSpeed(int speed) {
		this.speed = speed;
	}

	public int getSize() {
		return size;
	}

	public void setSize(int size) {
		this.size = size;
	}

}
```

javabyKiran
java | selenium | python

```
package com.javabykiran1;

public class Car extends Vehicle {
	int CC;
	int gears;
	String color;

	public String getColor() {
		return color;
	}

	public void setColor(String color) {
		this.color = color;
	}

	public int getCC() {
		return CC;
	}

	public void setCC(int cC) {
		CC = cC;
	}

	public int getGears() {
		return gears;
	}

	public void setGears(int gears) {
		this.gears = gears;
	}
}
```

javabyKiran
java | selenium | python

```
package com.javabykiran1;

public class Test {
	public static void main(String[] args) {
		Car b1 = new Car();
		b1.color = "red";
		b1.setSpeed(200);
		b1.setSize(22);
		b1.CC = 1000;
		b1.gears = 5;
		System.out.println("Color of Car : " + b1.color);
		System.out.println("Speed of Car : " + b1.getSpeed());
		System.out.println("Size of Car : " + b1.getSize());
		System.out.println("CC of Car : " + b1.CC);
		System.out.println("No of gears of Car : " + b1.gears);
	}
}
```

javabyKiran
java | selenium | python

- **Multilevel Inheritance**

Example #3

```
package com.jbk1;

public class X {
	public void methodX() {
		System.out.println("Class X method");
	}
}
```

```
package com.jbk1;

public class Y extends X {
	public void methodY() {
		System.out.println("class Y method");
	}
}
```

javabyKiran
java | selenium | python

```
package com.jbk1;

public class Z extends Y {
	public void methodZ() {
		System.out.println("class Z method");
	}

	public static void main(String args[]) {
		Z obj = new Z();
		obj.methodX(); // calling grand parent class method
		obj.methodY(); // calling parent class method
		obj.methodZ(); // calling local method
	}
}
```

Example #4

```
package com.jbk1;

class Car {
	public Car() {
		System.out.println("Constructor of class Car");
	}

	public void vehicleType() {
		System.out.println("Vehicle Type : Car");
	}
}

class Maruti extends Car {
	public Maruti() {
		System.out.println("Constructor of class Maruti");
	}

	public void brand() {
		System.out.println("Brand : Maruti");
	}

	public void speed() {
		System.out.println("Max speed: 90Kmph");
	}
}

public class Maruti800 extends Maruti {
	public Maruti800() {
		System.out.println("Constructor of class Maruti800");
	}

	public void speed() {
		System.out.println("Max speed: 80Kmph");
	}

	public static void main(String[] args) {
		Maruti800 obj = new Maruti800();
		obj.vehicleType();
		obj.brand();
		obj.speed();
	}
```

javabyKiran java | selenium | python

```
}
```

Example #5

```
package com.jbk1;

class User {
	String name;
	int age;
	long ph;
}

class Employee extends User {
	String specialization;
}

class Manager extends User {
	String department;
}

class Main {
	public static void main(String[] args) {
		Employee e1 = new Employee();
		e1.name = "Candid";
		e1.age = 22;
		e1.ph = 123456789l;
		e1.specialization = "Java";
		Manager m1 = new Manager();
		m1.name = "java";
		m1.age = 25;
		m1.ph = 345789l;
		m1.department = "HR";
		System.out.println(e1.name);
		System.out.println(e1.age);
		System.out.println(e1.ph);
		System.out.println(e1.specialization);
		System.out.println(m1.name);
		System.out.println(m1.age);
		System.out.println(m1.ph);
		System.out.println(m1.department);
```

javabyKiran java | selenium | python

```
        }
}
```

- **hierarchical inheritance**

Example #6

```
package com.jbk1;

class A {
        public void methodA() {
                System.out.println("method of Class A");
        }
}

class B extends A {
        public void methodB() {
                System.out.println("method of Class B");
        }
}

class C extends A {
        public void methodC() {
                System.out.println("method of Class C");
        }
}

class D extends A {
        public void methodD() {
                System.out.println("method of Class D");
        }
}

public class HierarchialEx {

        public static void main(String args[]) {
                B obj1 = new B();
                C obj2 = new C();
                D obj3 = new D();
                // All classes can access the method of class A
```

javabyKiran java | selenium | python

```
		obj1.methodA();
		obj2.methodA();
		obj3.methodA();
	}
}
```

Example #7

```
package com.jbk2;

class A {
	public void methodA() {
		System.out.println("method of Class A");
	}
}

class B extends A {
	public void methodB() {
		System.out.println("method of Class B");
	}
}

class C extends A {
	public void methodC() {
		System.out.println("method of Class C");
	}
}

class D extends A {
	public void methodD() {
		System.out.println("method of Class D");
	}
}

public class Test {
	public void methodB() {
		System.out.println("method of Class B");
	}
```

javabyKiran
java | selenium | python

```
	public static void main(String[] args) {
		B obj1 = new B();
		C obj2 = new C();
		D obj3 = new D();
		obj1.methodA();
		obj2.methodA();
		obj3.methodA();
	}
}
```

- **protected variable usage**
    - create 2 packages jbk1 and jbk2
    - protected member should be called in subclass by subclass object as shown
    - If we create object of super class, then protected member will not get called in sub class.

javabyKiran java | selenium | python

Example #8

```
package com.jbk2;

public class Shape {
	protected int sides;

	public Shape() {
		sides = 3;
	}

	public int getSides() {
		return sides;
	}

	public void printSides() {
		System.out.println("This object has " + sides + "     sides.");
	}
}
```

```
package com.jbk3;

import com.jbk2.Shape;

public class Square extends Shape {
	public Square(int nSides) {
		sides = nSides; // possible
		// don't need to call super class
		// constructor due to protected type of variable.
	}

	void display() {
		Shape shape= new Shape();
		System.out.println(shape.sides);//error as we are not calling
		// through object of subclass
	}
}
```

javabyKiran
java | selenium | python

```
package com.jbk2;

import com.jbk3.Square;

public class ProtectedVariableDemo {
	public static void main(String args[]) {
		Square sObj = new Square(10);
		sObj.printSides();
	}
}
```

Homework

- Solve test on jbktest.com for inheritance
- Read jbktutorials.com
    - https://www.jbktutorials.com/corejava/inheritance-in-java.php#gsc.tab=0
- Read interview questions
    - https://www.jbktutorials.com/core-java-interview-questions/inheritance-interview-questions-1.php#gsc.tab=0
    - https://www.jbktutorials.com/core-java-interview-questions/inheritance-interview-questions-2.php#gsc.tab=0

Download

https://drive.google.com/drive/folders/1AglbIcIe5bQNRzJW5HVRbOMCqilRY2af?usp=sharing

javabyKiran
java | selenium | python