//public: is a type of access modifiers (private, default, protected, public)

Access level: everywhere (within the class, outside the class, within the package, outside the package

**//A class is like an object constructor or a “blueprint” for creating objects**

//naming convention: camelCase (class: 1st letter🡪uppercase; 1st letter of subsequent words🡪uppercase)

**public class Car {**

//private: is a type of access modifiers (private, default, protected, public)

Access level: within the class

All instance variables should be declared as private

**//instance variable: special variable that hold the data item (attribute) of a class’s objects**

//declaring all variables🡪give a name (yearMade) & data type (int)

//int: the instance variable yearMade can only store integer values

//naming convention: camelCase (variable: 1st letter🡪lowercase; 1st letter of subsequent words🡪uppercase)

**private int yearMade;**

//public: any instance of class Car can have access to public methods and can invoke them

//void: specifies that this method should not have a return value

//setYearMade(): method (setter/mutator)

**Method: a group of Java statements that perform some operation on some data**

Setter/mutator: to reset the value of an instance variable

//(int newYearMade): method input/parameter

**Parameter: carry data into a method 🡪input**

For each parameter used by the method: list the parameter type (int) followed by the parameter name (newYearMade)

**public void setYearMade (int newYearMade){**

//if statement

**if (newYearMade>1950 && newYearMade<=2030){**

//assignment statement: designate a value for a variable

**yearMade=newYearMade;**

**}**

**}**

//public: any instance of class Car can have access to public methods and can invoke them

//int: specifies the return type of this method

If deleting int, it will show: invalid method declaration, return type required

//getYearMade():method(getter/accessor)

**Method: a group of Java statements that perform some operation on some data**

Getter/accessor: only retrieve the current value of an instance variable

//( ): parameter (empty because this method gets data out of a Car object, not data into Car object)

**public int getYearMade( ){**

//method output🡪return value

//return value: carry data out of a method back to the calling code

**return yearMade;**

**}**

**}**

//a driver class is used to test another class

**A driver class creates an instance of the Car class and invoke all its methods to see if they work correctly**

//naming convention: camelCase (class: 1st letter🡪uppercase; 1st letter of subsequent words🡪uppercase)

**public class ApplicationDriver{**

//public: access modifier (making the main( ) method globally available)

//static: keyword

Static means the main( ) method belongs to the class instead of the objects of the class

JVM can invoke the main( ) method without instantiating the class

//void: keyword (specifies that this method should not have a return value)

//main: name of Java main method

Main is an identifier not keyword

Main( ) method is the entry point of any Java program

It is a default signature which is predefined in JVM, it is called by JVM to execute a program line by line and end the execution after completion of this method

//(String [ ] args): array name: args[ ], data type: string)

**public static void main (String [ ] args){**

//instantiating an object

//instantiation statement (Car sedan;) & assignment statement (sedan=new Car( );) combined

//new: keyword (a Java operator that creates the object)

//Car: non-primitive data type (class)

//sedan: variable name (reference variable)

**Car sedan = new Car( );**

//dot(.) is used to access the object’s attributes and methods

**sedan.setYearMade(2021);**

//System.out,println: print the argument passed to it

//System: a final class in java.lang package

//out: is a static member field of System class and is of type PrintStream

//println: a method of PrintStream class

**System.out.println(“The car is made in ”+sedan.getYearMade( ));**

**}**

**}**