|  |  |
| --- | --- |
| **Skill 25.2: Exercise 1** | |
| The MyCar class below extends the Car class. For each line of code indicated with a letter (A – E), indicate whether the statement is valid or invalid. If it is invalid, indicate why. | |
| public abstract class Car{  private int year = 2015;  private String model = “Landcruiser”;  (A)  public abstract String getMake();  (B)  public abstract int getYear(){  return year;  }  (C)  public String model(){  return model;  }  } | public class MyCar extends Car{  public static void main(String args[]){  (D)  Car newCar = new Car();  }  (E)  public String getMake(){  return “Toyota”;  }  } |
| (A)  (B)  (C)  (D)  (E) | |

|  |  |
| --- | --- |
| **Skill 25.2: Exercise 2** | |
| 1. Declare an abstract class Insect. Then declare another class called Bee which inherits Insect. 2. Declare a method in the Insect class called getLegs(), which returns the number of legs as an int. 3. Declare a Boolean abstract method in the Insect class called canFly() 4. In the Bee class, implement the canFly method 5. Write the main class and in the main class instantiated a Bee object, then call getLegs() and canFly() | |
|  |  |
|  | |

|  |  |
| --- | --- |
| **Skill 25.4: Exercise 1** | |
| 1. Declare an interface called Animal 2. Declare a class called Ant that implements Animal | |
|  |  |

|  |  |
| --- | --- |
| **Skill 25.4: Exercise 2** | |
| Consider the animal interface below. The Unicorn and Dinosaur classes implement the Animal interface. Write the Unicorn and Dinosaur classes. | |
| public interface Animal {  //All the methods below are abstract  void setAge(int a);  void setType(String t);  boolean getEats(boolean e);  } | |
|  |  |