6. ABSTRACT CLASSES

1. Reason behind abstraction:

Some times the super class has no any meaningful implementation for any particular method. Such methd can be declared as an abstract so that later in future in subclass it may be get implemented. Abstract method has no any body. It has just signature.

2. Abstract Methods:

Syntax of declaring method as an abstract:

access modifier abstract returntype methodname(parameter list);

If atleast one method is declared as an abstract, the class must also declared as an abstract, But it is not vice versa.
When the class in declared as an absract it should get extended to override its abstract methods.
Abstract classes are known as **Partial Abstraction** as they may contain the concrete methods means the methods those have body. It may contain its own instance variables, constrcutors also.

Note: We can not create the instance of absract class directly. It will get instantiated via subclass

3. Properties of abstract class:

1. Abstract class may contain concrete methods i.e. methods with body.
2. Absract class may be blank.
3. The absract methods in absract class must be explicitly defined as an abstract.
4. The absract methods may be protected, default and public.
5. We can declared instance variables, static variables, final variables, and constructors in absract class.
6. We cant create instance of abstract class directly. It will get created throgh subclass. But we can create the reference of abstract class
7. All abstract methods of abstract class must be overriden in its subclass either, or the subclass must be declared as an abstract if it does not want to implement all abstract methods of its super class.
8. We can not declare absract class and abstract method as final.
9. A class can not extend more than one abstract class. Because java does not support multiple inheritance through classes.

Example:

```
abstract class Camera
```

```
{
  public void frontCamera()
        {
              //implementation of frontCamera method.
        }
   public abstract void  frontFlash(); // no implementation
}

class CameraWithFlash extends Camera
{
        // frontFlash should be overridden here

        public void frontFlash()  {
              //implementation for frontFlash method
        }
        public void zoomCamera()  {
              //implementation for own zoomCamera method
        }
        public static void main(String[] args) {

              Camera cm=new CameraWithFlash();  // abstract class reference variable
              cm.frontCamera();
              cm.frontFlash();

              CameraWithFlash cmf= (CameraWithFlash)cm; // down casting
              cmf.zoomCamera();
        }
}
```

Thus whener you need a partial abstraction go for abstract classses.

ASSIGNMENTS

1. Write a program to achieve partial abstraction in Vehical class.
2. Declare the Vehical class as an abstract and define some abstract methods in Vehical. Implement the code to override the abstract methods.