

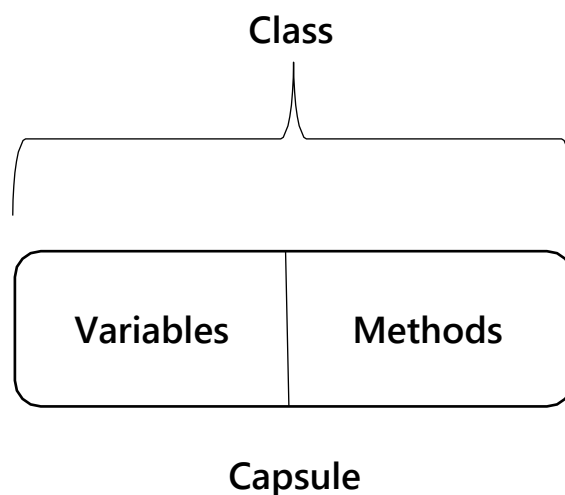
Encapsulation :-

What is encapsulation?

It is a mechanism through we can wrap the data members and member methods of class in a single unit, is called encapsulation.

- Declare the class variables as a private.
- Declare the class method as a public.

Ex. Class is the best example of encapsulation.



Abstraction :-

What is abstraction?

It is a process of hiding the implementation details from the user, only the highlighted set of services provided to the user.

Advantage :-

1. Security
2. Enhancement

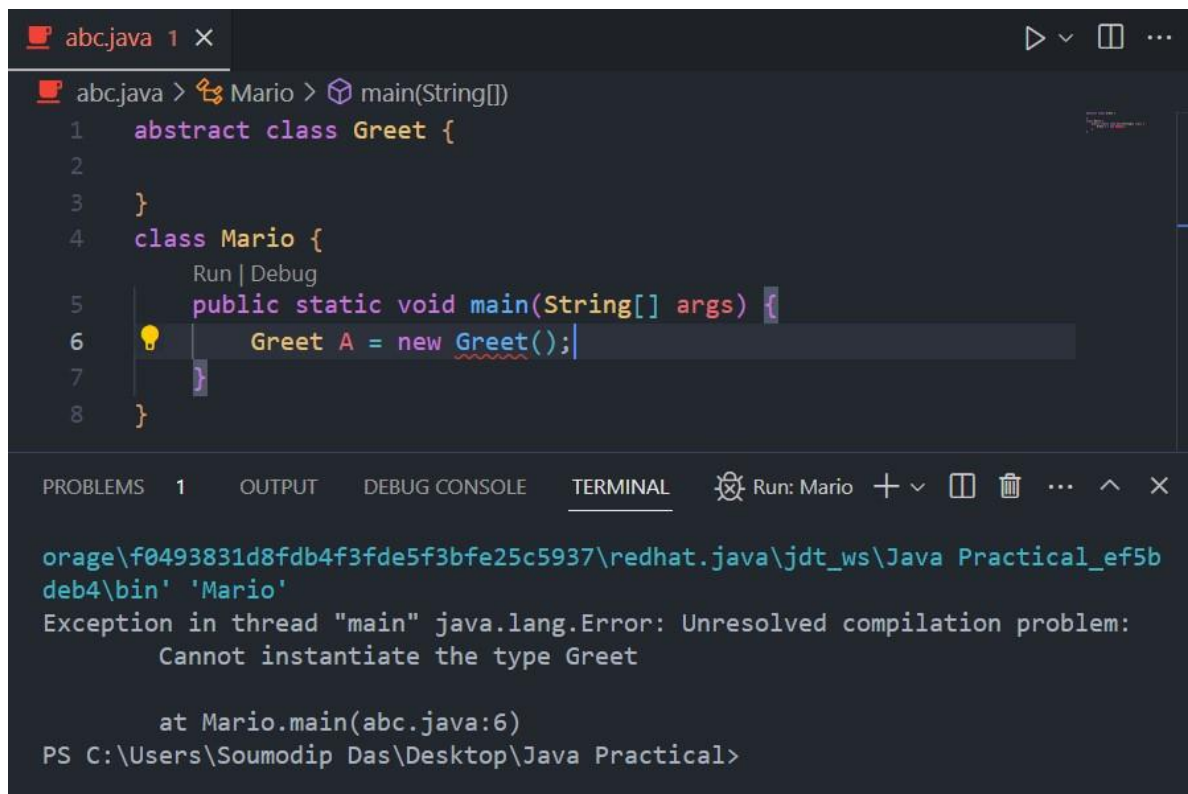
Abstraction

- Abstract class
- Interface

What is abstract class ?

A class which contains the abstract keyword in it's declaration is called abstract class.

1. We cannot create object for abstract class.



```
abc.java 1 x
abc.java > Mario > main(String[])
1  abstract class Greet {
2
3  }
4  class Mario {
5      Run | Debug
   public static void main(String[] args) {
6      Greet A = new Greet();
7  }
8  }

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL Run: Mario + - [ ] [ ] ... ^ X

orange\f0493831d8fdb4f3fde5f3bfe25c5937\redhat.java\jdt_ws\Java Practical_ef5b
deb4\bin' 'Mario'
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    Cannot instantiate the type Greet

    at Mario.main(abc.java:6)
PS C:\Users\Soumodip Das\Desktop\Java Practical>
```

But we can create reference of abstract class.

```
abc.java 1 X
abc.java > Luigi
1  abstract class Greet {
2
3  }
4  class Mario extends Greet {
5
6  }
7  class Luigi {
8      Run | Debug
9      public static void main(String[] args) {
10         Greet A = new Mario();
11     }
}
```

2. It may or may not contain abstract method.
3. It can have abstract & non-abstract methods.
4. To use an abstract class, we have to inherit it from sub classes.
5. If a class contain partial implementation then we should declare a class as abstract.

Ex.

```
abstract class Greet {
    abs void()
}
```

```
abs class Greet {
    public void {}
}
```

```
abstract class person {
    public abstract void write()
}
```

```
class student extends person{
}
```

```
class teacher extends person{
}
```

```
abc.java x
abc.java > Student > sleep()
4  class Teacher extends Person { //super class
5      public void sleep() {
6          System.out.println(x: "The time is 12:00 AM, teacher will sleep.");
7      }
8  }
9  class Student extends Person {
10     public void sleep() {
11         System.out.println(x: "The time is 2:00 AM, student will sleep.");
12     }
13 }
14 class School {
15     public static void main(String[] args) {
16         Teacher t = new Teacher();
17         Student s = new Student();
18         t.sleep();
19         s.sleep();
20     }
21 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

PS C:\Users\Soumodip Das\Desktop\Java Practical> & 'C:\Program Files\Eclipse Adoptium\jdk-17.0.6.10-hotspot\bin\java.exe' '-cp' 'C:\Users\Soumodip Das\AppData\Roaming\Code\User\workspaceStorage\f0493831d8fdb4f3fde5f3bfe25c5937\redhat.java1'

The time is 12:00 AM, teacher will sleep.
The time is 2:00 AM, student will sleep.
PS C:\Users\Soumodip Das\Desktop\Java Practical>

Abstract Method(Incomplete) :-

What is abstract method?

A method which contain abstract modifier at the time of declaration is called abstract method.

- It can be used in abstract class.
- It does not contain any body "{ }" and always ends with ";" .
- Abstract method must be overridden in a sub classes otherwise it will also become an abstract class.
- Whenever the action is common but implementation are different then we should abstract method.

```
class vehicle {  
    void wheels ();  
}
```

We can never say that how many wheels can exist in a vehicle, so we should define the method as abstract.

```
abstract class vehicle {  
    abstract void wheels ();  
}
```

var-args method

It is known as variable number of arguments method. By using this concept we can pass any number of parameters including zero parameter to the calling method.