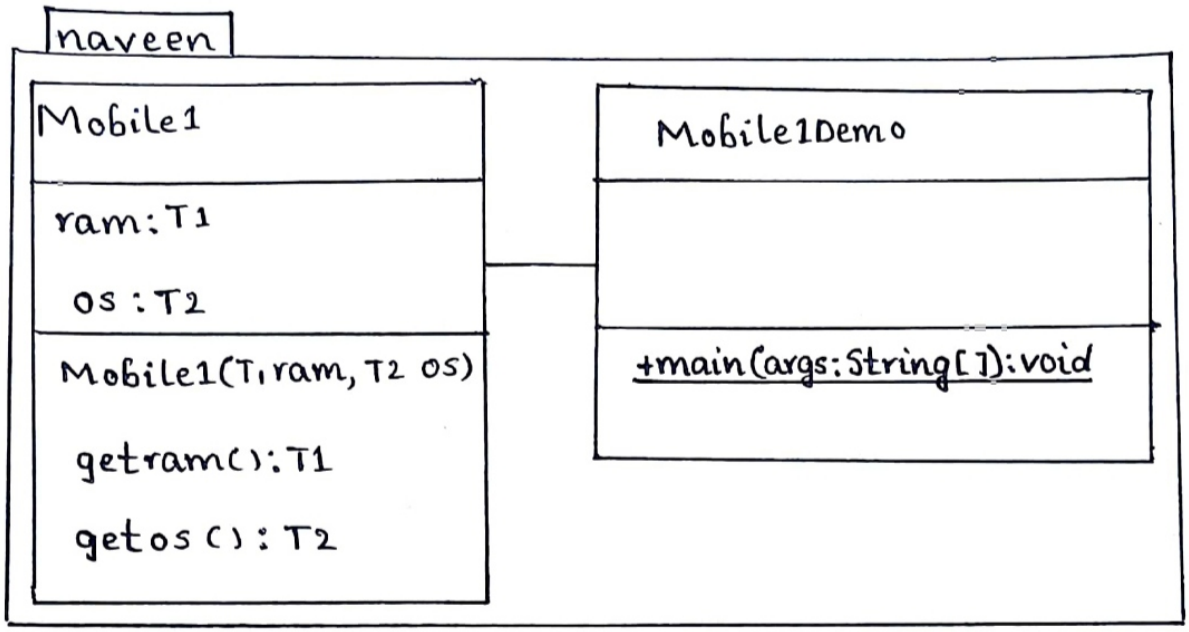
SKILLING EXERCISE-4

Name: Badisa Naveen

Reg.no: 2000031509

CLASS DIAGRAM:



Generic:

Generics mean parameterized types. The idea is to allows type (Integer, String ,…etc, and user-defined types) to be a parameter to methods, classes, and interfaces. Using Generics, it is possible to create classes that work with different data types. An entity such as class, interface, or method that operates on a parameterized type is called a generic entity.

Why Generics?

The **Object** is the superclass of all other classes and Object reference can refer to any type object. These features lack type safety. Generics add that type safety feature.
For example, classes like HashSet, Array List, HashMap, etc use generics very well. There are some fundamental differences between the two approaches to generic types.

Advantages of Generics: Programs that use Generics has got many benefits over non-generic code.

1. **Code Reuse**: We can write a method/class/interface once and use it for any type we want.
2. **Type Safety:**
   Generics make errors to appear compile time than at run time Generics make errors to appear compile time than at run time (It's always better to know problems in your code at compile time rather than making your code fail at run time).

Code:

```
package naveen;
public class Mobile1<T1,T2,T3,T4,T5> {
        T1 ram;
        T2 os;
        T3 rom;
        T4 price;
        T5 product_id;
    Mobile1(T1 ram,T2 os,T3 rom,T4 price,T5 product_id)
    {
        this.ram=ram;
        this.os=os;
        this.rom=rom;
        this.price=price;
        this.product_id=product_id;
    }
     T1 getram()
    {
        return this.ram;
    }
     T2 getos()
     {
         return this.os;
     }
     T3 getrom()
     {
         return this.rom;
     }
     T4 getprice()
     {
         return this.price=price;
     }
     T5 getproduct_id()
     {
         return this.product_id=product_id;
     }
```

```
}
package naveen;
public class Mobile1Demo {
        public static void main(String[] args) {
    Mobile1<Integer,String,Integer,Double,Integer> intobj = new
Mobile1<Integer,String,Integer,Double,Integer>(8,"Android",64,1500
0.35,20000315);
    System.out.println("Details of Mobile:");
    System.out.println("Product_id="+intobj.getproduct_id());
    System.out.println("Os="+intobj.getos());
    System.out.println("Ram="+intobj.getram());
    System.out.println("Rom="+intobj.getrom());
    System.out.println("Price="+intobj.getprice());
        }
}
```

ScreenShots: