Experiment 9: Dynamic Method Dispatch in Java

Aim

To write a Java program to demonstrate Dynamic Method Dispatch.

Theory:

Dynamic Method Dispatch is a process in which a method call is resolved at runtime rather than compile time. It is achieved using method overriding and inheritance, where a superclass reference refers to a subclass object. The method of the subclass is called based on the object type.

Algorithm

1. Create a superclass with a method
2. Create subclasses that override the method
3. Declare a superclass reference
4. Assign subclass objects to the reference
5. Call the overridden method

Program Code:

```
class MobileRecharge {

   void rechargeType() {
      System.out.println("General Mobile Recharge");
   }
}

class PrepaidRecharge extends MobileRecharge {
   void rechargeType() {
      System.out.println("This is a Prepaid Mobile Recharge");
   }
}

class PostpaidRecharge extends MobileRecharge {

   void rechargeType() {
      System.out.println("This is a Postpaid Mobile Recharge");
   }
}

class DynamicMethodDispatchDemo {
   public static void main(String[] args) {

      MobileRecharge recharge;   // Superclass reference

      recharge = new PrepaidRecharge();
      recharge.rechargeType();

      recharge = new PostpaidRecharge();
      recharge.rechargeType();
   }
}
```

Result

The program successfully demonstrates an abstract class. The abstract method is implemented in the subclass and executed.