Name- Pradnya Abhay Magennavar
Roll no- 12
Div-B **Batch-**B1
Experiment no-03

Experiment name- Implement java programs based on abstract and final keyword.

Java abstract Keyword

The abstract keyword is used to achieve abstraction in Java. It is a non-access modifier which is used to create abstract class and method.
The role of an abstract class is to contain abstract methods. However, it may also contain non-abstract methods. The method which is declared with abstract keyword and doesn't have any implementation is known as an abstract method.
Syntax:-

```
abstract class Employee
{
abstract void work();
}
```

Final keyword-
The **final keyword** in java is used to restrict the user. The java final keyword can be used in many context. Final can be:
variable
method
class
The final keyword can be applied with the variables, a final variable that have no value it is called blank final variable or uninitialized final variable. It can be initialized in the constructor only. The blank final variable can be static also which will be initialized in the static block only

1.Program for final variable.
Input-

```
class Main
{
  public static void main(String[] args)
  {
    final int AGE = 32;
    AGE = 45;
    System.out.println("Age:" +AGE);
  }
}
```

Output-

```
C:\Users\LENOVO-PC\Desktop>javac Main1.java
[
C:\Users\LENOVO-PC\Desktop>java Main1
SAge: 45
```

2. Java final method program.

Input-

```
class Bike {

   void run()

   {

      System.out.println("running");
```

```
    }
}


class Honda extends Bike
 {
    void run()
    {
        System.out.println("running safely with 100kmph");
    }


    public static void main(String args[])
   {
        Honda honda = new Honda();
        honda.run();
    }
}
```

Output-

```
C:\Users\LENOVO-PC\Desktop\New folder>javac Honda.java

C:\Users\LENOVO-PC\Desktop\New folder>java Honda
running safely with 100kmph
```

3.Abstract class containing abstract method.

Input-

```
abstract class Vehicle
{
    abstract void bike();


}
class Honda extends Vehicle
{
```

```
    @Override
    void bike()
   {
        System.out.println("Bike is running");


   }


}


class Example1
{


    public static void main(String[] args)
    {


    Honda obj=new Honda();
    obj.bike();
    }
}
```

Output-

```
C:\Users\LENOVO-PC\Desktop>javac  Example1.java

C:\Users\LENOVO-PC\Desktop>java  Example1
Bike is running
```