Scenario - 3 (Achiving Encapsulation)

- Encapsulation provide us control access.

- Private members are accessable with in the same class.

System.exit(0) :- It is used to Shut down the program.

- Encapsulation is providing security to the important data members of your class.

Example 1 (Using Hero)

```java
class Bank
{
    private int amt = 10000;
    int pin = 9871;
    public void setAmt (int p, int a)
    {
        if (pin == p)
        {
            amt = a;
        }
        else
        {
            System.out.println("Invalid PIN");
            System.exit(0);
        }
    }

    public int getAmt (int p)
    {
        if (pin != p)
        {
            System.out.println("Invalid PIN");
            System.exit(0);
        }
        return amt;
    }
}
```

```java
class Hero              JVM
{
    public static void main (String args[])
    {
        Bank b = new Bank();
        int p = 9871;
        b.setAmt (p, 5000);
        int a = b.getAmt (p);
        System.out.println("Amount ="+a);
    }
}
```

P	9871		P	9871
a	5000		a	5000

Output:-

```
Amount = 5000
```

Example 2 (Using Villain)

class Bank

```java
class Bank
{
   Private int amt = 10000;
      int pin = 9871;
   Public void setAmt (int p, int a)
   {
      if (pin == p)
      {
         amt = a;
      }
      else
      {
         System.out.println("Invalid PIN");
         System.exit(0);
      }
   }

   Public int getAmt (int p)
   {
      if (pin != p)
      {
         System.out.println("Invalid PIN");
         System.exit(0);
      }
      return amt;
   }
}
```

class Villain

```java
class Villain
{
   public static void main (String args)
   {
      Bank b = new Bank();
      int p = 1234;
      b.setAmt (p, 0);
      int a = b.getAmt (p);
      System.out.println ("Amount=" +a);
   }
}
```

JVM

P 1234   a 0

Output:-

Invalid PIN