

```

Role of static field in Object creation :
* In Java whenever we create an object, a separate copy of all the non static fields will be
  created for each object and every object is stored in the heap.
/Program
class TestObject {
    int x = 10;
}
public class TestObject {
    public static void main(String[] args) {
        TestObject t1 = new TestObject();
        System.out.println(t1.x); //11
        System.out.println(t1.x); //12
    }
}

Static Field or Class Level Variable :
* If we declare a variable inside class with static keyword then it is called static field.
* If we declare a variable inside class with static keyword then it is called static field.
* If we declare a variable inside class with static keyword then it is called static field.
* When the class file is compiled then automatically static field will be
  initialized.

Static field while creating an object :
* To access the static field which is not initialized, we can directly access with the help of
  class name.
* When we declare a static field then a single copy of static field will be created and then the
  same copy will be shared by all the objects so we can say static field is mostly used
  for shared data.

/Program
class TestObject {
    static int x = 10; //static field
    public static void main(String[] args) {
        TestObject t1 = new TestObject();
        System.out.println(t1.x); //11
        t1.x = 20;
        System.out.println(t1.x); //12
        System.out.println(TestObject.x); //13
    }
}

Q) As a developer when we should declare a field as a non static field and when we should go with
Non Static Field :
* If the value of the variable is different with respect to object then we should declare non
  static field.
Static Field :
* Value of the variable is common for all the objects then we should declare
  static field.
* Example :
  class BankCustomer {
      int rollNumber;
      String name;
      String address;
      String phone;
      static String courseName = "NET";
      static String IFCODE = "009H000012";
      static String branchName = "Bihar";
  }

How to describe the use of static and non static field :
/Program
public class BankCustomer {
    int rollNumber;
    String name;
    String address;
    String phone;
    static String courseName = "NET";
    static String IFCODE = "009H000012";
    static String branchName = "Bihar";
}

public void setRollNumber(int roll, String name, String address) {
    this.rollNumber = roll;
    this.name = name;
    this.address = address;
}

public void getCustomerData() {
    System.out.println("Student Roll number is :" + rollNumber);
    System.out.println("Student Name is :" + name);
    System.out.println("Student Address is :" + address);
    System.out.println("Student Phone is :" + phone);
    System.out.println("Course Name is :" + courseName);
}

package com.ravindra;

public class UnderGraduate {
    public void print() {
        Student r1 = new Student();
        r1.setRollNumber(100, "Amit", "Kharagpur");
        r1.setAddress("Kharagpur");
        Student r2 = new Student();
        r2.setRollNumber(101, "Rahul", "Kharagpur");
        r2.setAddress("Kharagpur");
        System.out.println("Balance after deposit : " + balance);
    }
}

**** Data Hiding :
Data hiding is nothing but declaring our non static fields with private access modifier so our data will
not be accessible from outside the class or object of class which means no one can access our data
directly from outside the class.
* We can access the property of our data through methods so we can perform VALIDATION ON
  DATA which are coming from outer world.

package com.ravindra;

public class Current {
    private int balance = 1000; //Data hiding
    public void withdraw(int amount) {
        if(amount > balance) {
            System.out.println("Insufficient Balance");
            System.exit(0);
        } else {
            balance = balance - amount;
            System.out.println("Balance after withdraw : " + balance);
        }
    }
    public void checkBalance() {
        System.out.println("Current Balance is : " + balance);
    }
}

package com.ravindra;

public class BankCustomer {
    public static void main(String[] args) {
        Customer r1 = new Customer();
        r1.setRollNumber(100, "Amit", "Kharagpur");
        r1.setAddress("Kharagpur");
        Customer r2 = new Customer();
        r2.setRollNumber(101, "Rahul", "Kharagpur");
        r2.setAddress("Kharagpur");
        System.out.println("Current Balance is : " + checkBalance());
    }
}

java@javashaper:~/gcpad$
```

- How many ways we can initialize the object properties :
- 1) Using constructor with parameter (highlighted in red)
 - 2) By using method without parameter (Customer class)
 - 3) By using static method (BankCustomer class)
 - 4) Non static field initialization at the time of declaration