Why compiler automatically provides default no argument constructor to a class :

```
public class Student
{
  int roll;
  String name;
  public Student(){//default no argument constructor
  }
  public void show()
  {
    System.out.println("-"+name);
  }
}
public class Main
{
  public static void main(String [] args)
  {
    Student ref = new Student();
  }
}
```

Work of new keyword :

* It will allocate the memory for all the non static member like non static field and non static method
* It will initialize the default value for all the non static fields with default values.
  This process is known as **"Initialization Process"**.

* new keyword is responsible for providing the default value for the non static fields with the help of java compiler. This process is known as "Initialization Process".

As we know only class level variables are having default values so,new keyword is responsible to provide default value for the non static fields with the help of Java compiler.

Data type : Default value

```
byte : 0
short : 0
int : 0
long : 0L
float : 0.0f
double : 0.0D
char : '\uoo00'
boolean : false
String : null
Object : null (For any class i.e reference variable the default value is null)
```

Program that describes new keyword is responsible to allocate
the default values for non static variables :

```
package com.ravi.abc;

public class Student
{
  int roll;
  String name;
  double fees;
  char grade;

  public void show()
  {
      System.out.println(roll);
      System.out.println(name);
      System.out.println(fees);
      System.out.println(grade);
  }


}

package com.ravi.xyz;

import com.ravi.abc.Student;

public class Main
{
  public static void main(String[] args)
  {
      Student ref = new Student();
      ref.show();
  }

}
```

Initializing the non static fields with parameter variable as per requirement :

```
package com.ravi.hr;

public class Employee
{
  int employeeId;
  double employeeSalary;
  double employeeTerr;
  char employeeGrade;

  public void setEmployeeDetail(int id, String name, double salary)
  {
      employeeId = id;
      employeeName = name;
      employeeSalary = salary;
  }

  public void printEmployeeDetail()
  {
      System.out.println("Employee Id is :"+employeeId);
      System.out.println("Employee Name is :"+employeeName);
      System.out.println("Employee Salary is :"+employeeSalary);
      System.out.println("Employee Grade is :"+employeeGrade);
  }


  public void setTerritoryGrade()
  {
      if(employeeSalary >=5000)
      {
          employeeGrade = 'A';
      }
      else if(employeeSalary >=3000)
      {
          employeeGrade = 'B';
      }
      else if(employeeSalary >=1000)
      {
          employeeGrade = 'C';
      }
      else
      {
          employeeGrade = 'D';
      }
  }

}


package com.ravi.xyz;

import com.ravi.hr.Employee;

public class EmployeeDemo
{
    public static void main(String[] args)
    {
        Employee scott = new Employee();
        scott.setEmployeeDetail(55, "Scott", 5000);
        scott.setTerritoryGrade();
        scott.getEmployeeDetail();

        System.out.println("-----------");

        Employee smith = new Employee();
        smith.setEmployeeDetail(66, "Smith", 15000);
        smith.setTerritoryGrade();
        smith.getEmployeeDetail();

    }

}
```