Objectives

Click to Continue

This session will help you to understand the following,

- How to create custom objects?
- How to add custom objects inside collections?

How to store user defined objects in collections?

Click to Continue

User defined classes can be stored in any of the collections like list , set, & map.

This is a very important implementation a developer should learn. Application development uses this implementation extensively.

In the next try it out , we will see how user defined objects can be added and retrieved from an *ArrayList*. Same procedure applies for all other collections.

IMPORTANT: You should know how to create value or transfer objects taught in Java basics course.

Try it out – Adding user defined object in collection

Click to Continue

Consider a scenario in which you want to send the details of students to a method. The student details includes roll number , name , address, phone number and email id. Suppose there are 50 students what can be done to pass the details of all the 50 students together ?

Let us see how to solve it?

Step 1 : Create a Student class with the roll number, name , address as it's fields.

Step 2 : Create an object of Student class for each student and load the details.

Step 3 : Create an ArrayList and add each of the Student objects to the list.

Step 4 : Pass the list as argument to the method.

Try it out – Adding user defined object in collection

Click to Continue

Let us create a ***StudentManager*** class with method ***printStudentDetails()*** accepts the student details as a list of student objects.

Components to be developed:

1. **Student Class :** For holding the student details.
2. **StudentManager class :** Contains method for printing the student details.
3. **MainClass class :** Creates 5 student objects , adds them to a list and pass it to the *printStudentDetails()* method of the *StudentManager* class.

Try it out : Create Student class

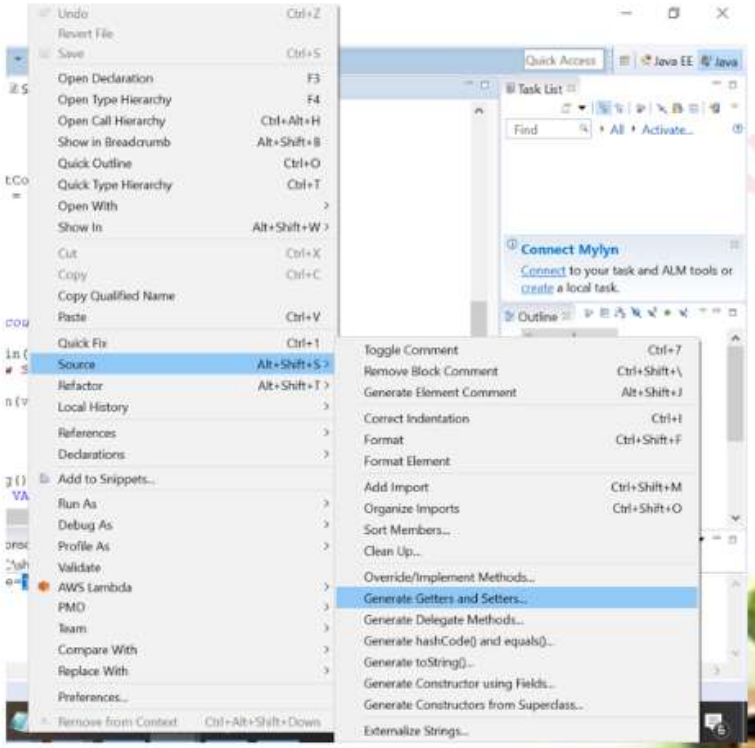
Click to Continue

The Student class should have the following fields to hold the student details

1. roll number
2. name
3. address
4. phone number
5. email id

All the fields should be created as private and accessed using public methods.

Create the student class, add all the required variables. Right click the source file and perform as given in the illustration.



Try it out : Student class Code

Click to Continue

```
public class Student {
    int rollNumber;
    String name;
    String address;
    long  phoneNumber;
    String emailId;
    public int getRollNumber() {
        return rollNumber;
    }
    public void setRollNumber(int rollNumber) {
        this.rollNumber = rollNumber;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    public long getPhoneNumber() {
        return phoneNumber;
    }
    public void setPhoneNumber(long phoneNumber) {
        this.phoneNumber = phoneNumber;
    }
    public String getEmailId() {
        return emailId;
    }
    public void setEmailId(String emailId) {
        this.emailId = emailId;
    }
```

All the methods starting with the word “get” are used to read the associated field value.

Example : *getName()* returns the name.

You can also add setter methods.

Try it out : Create StudentManager class Code

Click to Continue

Add **printStudentdetails** method for printing the student details.

The printStudentDetails() method accepts list of students as argument.

```
import java.util.List;

public class StudentManager {

    public void printStudentDetails(List<Student> students)

    {
        for (Student s: students)
        {
            System.out.println("Name:"+s.getName());
            System.out.println("Roll Number:"+s.getRollNumber());
            System.out.println("Email id:"+s.getEmailId());
        }

    }
}
```

Try it out : Create student Main class

Click to Continue

Create the main class , add two student objects in the list as mentioned below.

```
import java.util.ArrayList;
import java.util.List;

public class StudentMain {
    public static void main(String[] args) {
        List<Student> list = new ArrayList<Student>();
        Student s1 = new Student();
        s1.setName("John");
        s1.setRollNumber(1234);
        s1.setEmailId("John@abc.com");

        Student s2 = new Student();
        s2.setName("Tim");
        s2.setRollNumber(1235);
        s2.setEmailId("Tim@abc.com");

        list.add(s1);
        list.add(s2);

        StudentManager m = new StudentManager();
        m.printStudentDetails(list);
    }
}
```