Tutorial 11

Task 1:- With the same classes that you have developed in lecture, answer the following questions:

Ans:

Code	What it does	What type does it return
Student aStudent = new Student();	Creates a new Student object with default constructor values.	Student
Student anotherStudent = new Student("123","Mary donald);	Creates a new Student object with the provided ID and name.	Student
Student LastStudent = new Student("h213","Danielle Smith");	Creates a new Student object with the provided ID and name.	Student
aStudent.getName();	Gets the name of the aStudent object.	String
Course OODP= new Course("OODP","Object Oriented Programming and Design");	Creates a new Course object with the provided course name and course number.	Course
OODP.addStudent(aStudent);	Adds the aStudent object to the course.	void
OODP.addStudent(anotherStudent);	Adds the anotherStudent object to the course.	void
OODP.addStudent(LastStudent);	Adds the LastStudent object to the course.	void
OODP.numberOfStudents();	Gets the number of students enrolled in the course.	int
OODP.getStudentAt(2);	Gets the Student object at index 2 (third position) in the course.	Student
OODP.getStudentAt(2).getName();	Gets the name of the Student object at index 2 in the course.	string
OODP.getStudentAt(2).getName().charAt(0);	Gets the first character of the name of the Student object at index 2 in the course.	char

Task 2:- In the course class that you have developed, change the array to arrayList and update the class so that it does all the tasks with arrayList instead of array. (Update constructor, addStudent() and getStudent() methods)

Ans:

```
package week11;

import java.util.ArrayList;


public class Course {
```

```
private int maximumEnrolment = 50;
private String courseName;
private String courseNumber;
private ArrayList<Student> students = new ArrayList<>();
private int numEnrolled = 0;

// Default constructor
public Course() {
}

// Parameterized constructor
public Course(String courseName, String courseNumber) {
   this.courseName = courseName;
   this.courseNumber = courseNumber;
}

// Getters for courseName and courseNumber
public String getCourseName() {
   return courseName;
}

public String getCourseNumber() {
   return courseNumber;
}

// Setters for courseName and courseNumber
public void setCourseName(String courseName) {
   this.courseName = courseName;
}

public void setCourseNumber(String courseNumber) {
   this.courseNumber = courseNumber;
}
```

```
    public void addStudent(Student aStudent) {
        if (numEnrolled < maximumEnrolment) {
            students.add(aStudent);
            numEnrolled++;
        } else {
            System.out.println("Maximum enrollment reached. Cannot add more students.");
        }
    }

    public Student getStudentAt(int anIndex) {
        if (anIndex >= 0 && anIndex < numEnrolled) {
            return students.get(anIndex);
        } else {
            System.out.println("Invalid index.");
            return null;
        }
    }

    public int numberOfStudents() {
        return numEnrolled;
    }

    public String toString() {
        return courseName + " " + courseNumber + " has " + numberOfStudents() + " students";
    }
}
```