ESc101 : Fundamental of Computing

I Semester 2008-09

Lecture 22

Object Oriented programming

- Method Overloading in Java.
- Constructor
- Building complex classes using Point class

Circle, Triangle,...

Method overloading in JAVA

Question : In a class, can there be two methods with the same name?

Method overloading in JAVA

Question : In a class, can there be two methods with the same name?

Yes, but they should differ in the their **signatures**

Signature of method

return_type method_name (type1, type2, ...)

For example,

• public static int absolute(int i) has signature

int absolute(int)

• public static double absolute(int i, double d) has signature double absolute (int, double)

Note : The order matters in the parameter list.

Constructor

Constructor : a method with the same name as that of the class.

- Each class has a default constructor without parameters. For example, Point() constructor for class Point
- It has **No** return type.
- When is it executed ? : Immediately after the attributes of the newly created object have been initialized their (default) initial values.
- There may be more than one constructor (due to availability of method overloading)



Constructor

NOTE : The default constructor is no more available once we define one or more constructors. So we may have to define the default constructor explicitly if we need it.

Example

```
public class Point
{
   double x;
   double y;
   Point(double x1, double y1)
      x = x1;
    {
      y = y1;
    }
   Point() { x=0; y=0; }
   public double distance_from_origin()
       return(Math.sqrt(x*x + y*y)); }
    {
```

One more method for class Point

we want to add one more method to the class to compute distance of the point from another point.

One more method for class Point

we want to add one more method to the class to compute distance of the point from another point.

public double distance_from_point(??????)

One more method for class Point

we want to add one more method to the class to compute distance of the point from another point.

public double distance_from_point(Point Q)

The implementation is given in Point.java file

Forming complex data types using Point class

We can use Points to define classes for

- 1. Triangle
- 2. Circle
- 3. Polygon
- 4. Line segment
- 5. Tetrahedron

Example 1 : class of Triangle

Class of triangle : overview

```
public class Triangle
{ Point P; Point Q; Point R;
```

Triangle(Point A, Point B, Point C)
// constructor

Triangle(double x1, double y1,..., double y3)
// constructor

```
public double perimeter()
//perimeter of triangle
```

public void translate(double x_diff, y_diff)

Class of triangle : actual implementation

Given in file Triangle.java

Example 2 : class of Circle

Class of Circle : overview

public class Circle

```
{ Point center; double radius;
```

Circle(Point P, double r) //Constructor

```
Circle()
//Constructor
```

```
public double perimeter()
// method for computing perimeter
```

```
public void translate()
// method fo translating circle
```

Class of Circle : actual implementation

Given in file Circle.java

Two problems to be solved

- 1. Given the coordinates of vertices of triangle, find the area of the triangle.
- 2. Given two circles with their center and radius, determine if they intersect.

Solution of problems using OOP

Problem1 : Given the coordinates of vertices of triangle, find the area of the triangle.

Given in triangle_program1.java file

Problem2 : Given two circles with their center and radius, determine if they Given in circle_program1.java file

The solution based on structured programming

The solution based on structured programming

- Adhoc
- Difficult to read
- repetition of methods

For examples, for computing distance between points

The solution based on Object Oriented Programming

- Easier to code
- Easier to understand
- avoids repetitions of code

The code for computing distance between points is available from class Point itself, so no need to design it again in the two programs

Home work :

 Include the methods Area() of triangle_program1.java in the class Triangle itself.

Rewrite program for problem 1 based on the modified Triangle class

Include the method Intersect_circles() of circle_program1.java in the class
 Circle itself.

Rewrite program for problem 1 based on the modified Circle class