More about Inheritance & Polymorphism

COMP2026

PROBLEM SOLVING USING OBJECT ORIENTED PROGRAMMING

Overview

- Upcasting and Downcasting

Consider the following inheritance hierarchy

```
public class Shape
{
    private String name;
    public Shape(String name){
        this.name = name;
    }
    public String getName(){
        return name;
    }
}
```

```
public class Square extends Shape
{
    private double side;
    public Square(String name, double side){
        super(name);
        this.side = side;
    }
    public double getSide(){
        return side;
    }
}
```

```
public class Circle extends Shape
{
    private double radius;
    public Circle(String name, double radius){
        super(name);
        this.radius = radius;
    }
    public double getRadius(){
        return radius;
    }
}
```

What's wrong??

```
public class Shape
{
    private String name;

    public Shape(String name)
    {
        this.name = name;
    }

    public String getName()
    {
        return name;
    }
}
```

```
public class MainClass
{
    public static void main(String[] args)
    {
        Shape c = new Circle("MyCircle", 5);
        double r = c.getRadius();
        System.out.println(r);
    }
}
```

Error: Cannot find symbol – method getRadius()

```
public class Circle extends Shape
{
    private double radius;

    public Circle(String name, double radius)
    {
        super(name);
        this.radius = radius;
    }

    public double getRadius()
    {
        return radius;
    }
}
```

How to fix it?

What's wrong??

```
public class Shape
{
    private String name;

    public Shape(String name)
    {
        this.name = name;
    }

    public String getName()
    {
        return name;
    }
}
```

```
public class MainClass
{
    public static void main(String[] args)
    {
        Shape c = new Circle("MyCircle", 5);
        double r = ((Circle)c).getRadius();
        System.out.println(r);
    }
}
```

Downcasting to Circle

```
public class Circle extends Shape
{
    private double radius;

    public Circle(String name, double radius)
    {
        super(name);
        this.radius = radius;
    }

    public double getRadius()
    {
        return radius;
    }
}
```

Operator: instanceof

```
Shape c = new ...

if (c instanceOf Circle){
    double r = ((Circle)c).getRadius();

}
```

Part A
Discovery Exercises

Type your answer in **XXXXXXXX_lab10.docx**

Part B
Programming Exercises

Lab Exercise Submission

- Submit the following to Moodle
  - ***XXXXXXXX*_ lab11.docx**
  - ***XXXXXXXX*_lab11.zip**

*Replace "***XXXXXXXX***" with your **student ID**

Deadline: Before next Monday noon

References

- Dean, J., & Dean, R. (2008). Introduction to programming with Java: A problem solving approach. Boston: McGraw-Hill.
- Forouzan, B. A., & Gilberg, R. F. (2007). Computer science: A structured programming approach using C (3rd ed.). Boston, MA: Thomson Course Technology.
- Gaddis, T. (2016). Starting out with Java (6th ed.). Pearson.
- Liang, Y. D. (2013). *Introduction to Java programming: Comprehensive version*. (8th ed.). Pearson.
- Schildt, H. (2006). *Java a beginner's guide*. New York: McGraw Hill.
- Wu, C. T. (2010). *An introduction to object-oriented programming with Java*. Boston: McGraw Hill Higher Education
- Xavier, C. (2011). Java programming: A practical approach. New Delhi: Tata McGraw Hill.
- Zakhour, S., Kannan, S., & Gallardo, R. (2013). The Java tutorial: A short course on the basics (5th ed.).
- yet another insignificant Programming Notes. (n.d.). Retrieved from https://www3.ntu.edu.sg/home/ehchua/programming