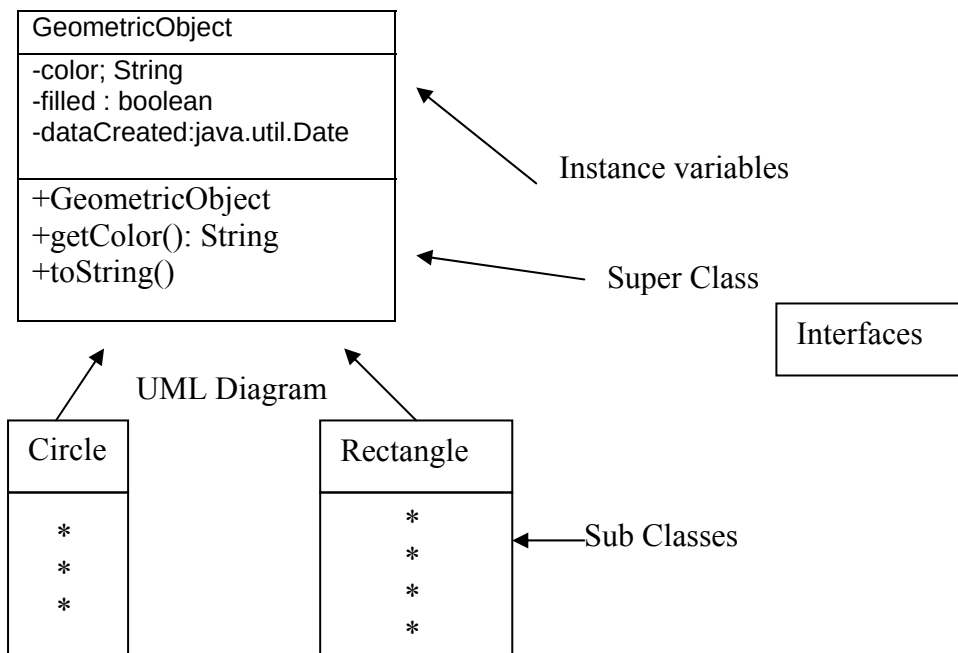
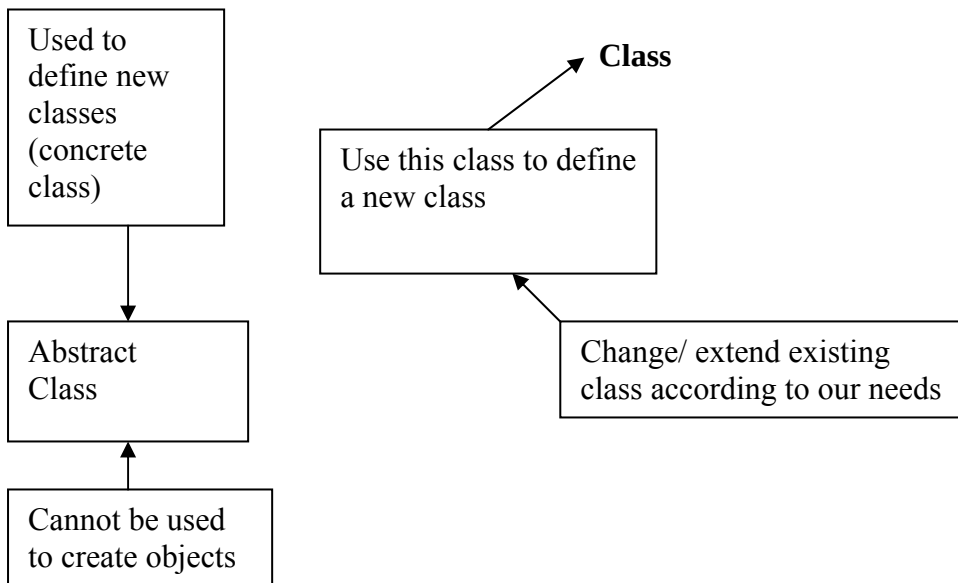
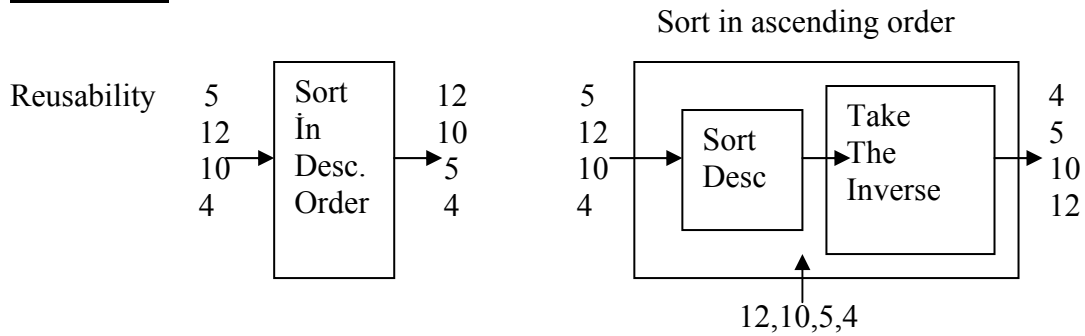


## Inheritance



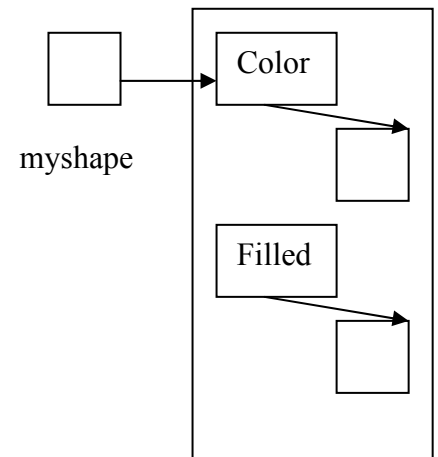
```

public class Geometric Object {
    private String color="white";
    private Boolean filled;
    private java.util.Date date Created;

```

Instance variables

private Date dataCreated;



```

    public GeometricObject(){// default constructor
        dataCreated=new java.util.Date();
        *
        * (getColor,setColoar etc.)
    }

    public void setFilled (boolean filled){
        this.filled=filled; }
}

```

```

public class Circle extends GeometricObject
{ private double radius;

    public Circle(double radius)
    { this.radius = radius;
    }

    public string toString()
    { return("circle radius is " + r);
    }

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

```

```

public class Rectangle extends GeometricObject
{ private double length, width;

    public Rectangle()
    { length = 1.0;
      width = 1.0;
    }

    public Rectangle(double length, double width)
    { this.length = length;
      this.length = width;
    }

    public String toString()
    { return("rectangle has width : " + width + "and has length : " +
length);
    }
}

```

toString() of rectangle overrides  
toString() of GeometricObject.

```

public static void main(String [] args)
{ GeometricObject [] myArray = new GeometricObject[10];
  Circle myCircle = new Circle(20);
  Rectangle myRectangle = new Rectangle(10, 20);
  Circle herCircle = new Circle(50);
  herCircle.setColor("purple");
  myArray[0] = myCircle;
  myArray[1] = myRectangle;
}

```

```

myArray[2] = herCircle;
for(int i = 0; i<3; i++)
{ myArray[i].toString;
}

```

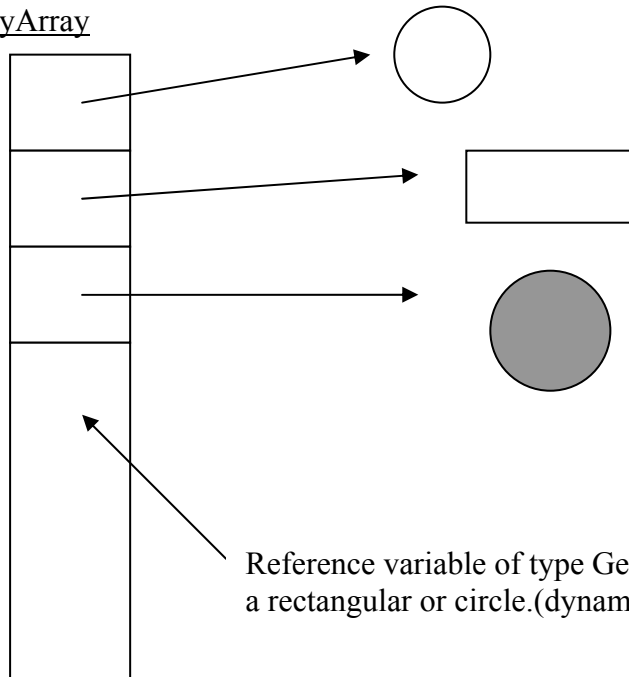
toString method is determined during execution time.

```

}

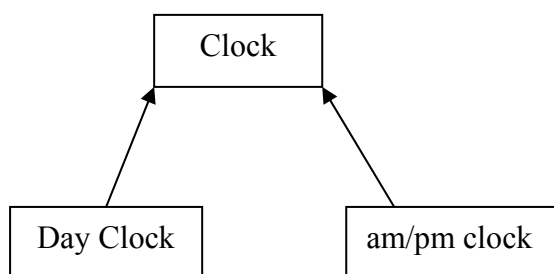
```

myArray



Reference variable of type GeometricObject it can refer a rectangular or circle.(dynamic binding)

CLOCK



I-am/pm clock:

```

public class AmPmClock extends Clock
{ private String brandName;

  public AmPmClock()
  { super();

```

```

    brandName = "Casio";
}

public AmPmClock(int h, int m, int s, String bName)
{ super(h,m,s);
  brandName = bName;
}

public void display() // overrides the display() method defined in clock
{ String hzero, mzero, szero;
  int h = super.getHours();
  int m = super.getMinutes();
  int s = super.getSeconds();
  if (h<12)
  { super.display();
    System.out.println("am");
  }

  else //afternoon
  { h = h-12;
    h = (h==0)? 12:h;
    hzero = (h<10)? "0":"";
    mzero = (m<10)? "0":"";
    szero = (s<10)? "0":"";
    System.out.println("Time is: " + hzero +h+":" + mzero +m+ ":" + szero
+s+ "pm");
  } // end of else

  System.out.println(""+brandName);
} // end of AmPmClock

```

## II-Day Clock:

```

public class AmPmClock
{ private String brandName;

  public AmPmClock()
  { super();
    brandName = "Casio";
  }

  public AmPmClock(int h, int m, int s, String bName)
  { super(h,m,s);
    brandName = bName;
  }

  public void display() // overrides the display() method defined in clock
  { String hzero, mzero, szero;
    int h = super.getHours();
    int m = super.getMinutes();
    int s = super.getSeconds();
    if (h<12)
    { super.display();
      System.out.println("am");
    }

    else //afternoon
    { h = h-12;
      h = (h==0)? 12:h;
      hzero = (h<10)? "0":"";
      mzero = (m<10)? "0":"";
      szero = (s<10)? "0":"";
      System.out.println("Time is: " + hzero + h + ":" + mzero + m + ":" +
szero + s + "pm");
    } // end of else

    System.out.println(""+brandName);
  }
}

```

```

} // end of AmPmClock

public class DayClock extends Clock
{ private String day;

    public DayClock()
    { super();
      day = "Sunday";
    }

    public DayClock(int h, int m, int s, String d)
    { super(h,m,s);
      day = d;
    }

    public void tick()
    { super.tick();
      if (get.Hours()==0 && get.Minutes==0 && get.Seconds==0)
      { if (day == "Sunday")
        day = "Monday";

        elseif (day == "Monday")
          day = "Tuesday";

        elseif (day == "Tuesday")
          day = "Wednesday";

        elseif (day == "Wednesday")
          day = "Thursday";

        elseif (day == "Thursday");
          day = "Friday";

        elseif (day == "Friday")
          day = "Saturday";

        else
          day = "Sunday";
      } // end of tick

    public void display()
    { super.display();
      System.out.println(""+day);
    }

    public String toString()
    { return(super.toString() + "" + day);
    }

} // end of DayClock

public static void main(String [] args)
{ Clock c = new Clock(0,0,0);
  DayClock dc = new DayClock(15,5,0, "Thursday");
  AmPmClock amPmC = new AmPmClock(5,7,0, "Rolex");
  dc.display();
  System.out.println("AmPm Clock is : " + amPmC);
  c=dc;
  c.tick();
  c.display();
} // end of main method

```

