

# Polimorfizam

- ❖ Objekti klase *ClockWithDate* moraju drugačije da reaguju na
  - operaciju *tick*, jer moraju da paze i na ažuriranje datuma
  - operaciju *getTime*, jer želimo da vrate i datum i vreme
- ❖ Funkcija članica koja će u izvedenim klasama imati *drugačiju implementaciju* deklariše se u osnovnoj klasi kao *virtuelna* funkcija (*virtual*). Izvedena klasa može da dâ svoju definiciju virtuelne funkcije, ali i ne mora:

```
class Clock {  
public:  
    Clock (int hh, int mm, int ss);  
  
    virtual void tick (); // Tick a second  
    virtual string getTime (); // Returns current time  
    void setTime (int hh, int mm, int ss);  
  
    int getHour() { return h; }  
    int getMinute() { return m; }  
    int getSecond() { return s; }  
    bool isMidnight () { return (h==0 && m==0 && s==0;) }  
  
private:  
    int h, m, s; // Hour, minute, second  
};
```

Virtuelne funkcije članice

---

# Polimorfizam

---

- ❖ Izvedena klasa može da dâ svoju definiciju virtuelne funkcije, tj. da je *redefiniše*, ali i ne mora (onda nasleđuje implementaciju):

```
class ClockWithDate : public Clock {
public:
    ClockWithDate (int y, int m, int d, int h, int min, int s);

    virtual void tick ();
    virtual string getTime ();

    string getDate ();
    void setDate (int yy, int mm, int dd);

private:
    int y, m, d;
};

void ClockWithDate::tick () {
    Clock::tick();
    if (isMidnight()) {
        // increment the date
    }
}

string ClockWithDate::getTime () {
    return to_string(m) + "/" + to_string(d) + "/" +
           to_string(y) + " " + Clock::getTime();
}
```