



Time STRUCTURE (2)

Pass-By-Pointer

Pass-By-Reference



MODIFY TIME (I)

- Update print to use pass-by-reference
- Add read
 - Return void
 - Pass-by-pointer
 - Prompts for and reads hours, minutes, and seconds
- `T data;`
- `function(T p) { ... }`
 - `function(data);`
- `function(T* p) { ... }`
 - `function(&data);`
- `function(T& p) { ... }`
 - `function(data);`



STRUCTURE SPECIFICATION AND PROTOTYPES

```
struct Time
{
    int    hours;
    int    minutes;
    int    seconds;
};

Time make_time(int h, int m, int s);
Time make_time(int s);
Time add(Time t1, Time t2);
```

```
void print(Time t);

void print(Time& t);
void read(Time* t);

// alternate versions
void print(Time* t);
void read(Time& t);
```



I/O FUNCTIONS

```
void print(Time& t)
{
    cout.fill('0');
    cout << t.hours << ":" <
        setw(2) << t.minutes <<
        ":" << setw(2) <<
        t.seconds << endl;
    cout.fill(' ');
}
```

```
void read(Time* t)
{
    cout << "Hours: ";
    cin >> t->hours;

    cout << "Minutes: ";
    cin >> t->minutes;

    cout << "Seconds: ";
    cin >> t->seconds;
}
```



Time DRIVER

Uses hard-coded test values and
arbitrary variable names

```
int main()
{
    Time t = make_time(3666);
    Time t;
    read(&t);
    print(t);

    Time s = make_time(1, 30, 4);
    Time s;
    read(&s);
    print(s);

    Time u = add(t, s);
    print(u);

    return 0;
}
```