Objects As Values In A Priority Queue, Reading

Just as with associative arrays, there are considerations with storing objects in priority queues. But there's one difference pertaining to **typename K**:

1. it either has no constructors *or* it has a default constructor (just as before),
2. it supports ***less-than*** comparisons (instead of equals-equals).

So once again, using **struct Time** as an example:

```
struct Time
{
  int hour, minute;
  Time(int h=0, int m=0){hour = h; minute = m;}
};
bool operator<(const Time& a, const Time& b) {return 60 * a.hour + a.minute < 60 * b.hour + b.minute;}
```

That's for the default hi-to-lo priority queue. For **lo-to-hi**, you simply *reverse the logic* for the less-than operator:

```
struct Time
{
  int hour, minute;
  Time(int h=0, int m=0){hour = h; minute = m;}
};
bool operator<(const Time& a, const Time& b) {return 60 * b.hour + b.minute < 60 * a.hour + a.minute;}
```

...or something equivalent.