

Second semester (OO) programming course @ University of Oslo, Norway

by Stein Gjessing: Objects with invariants and one vs. several users, **version 1**

One user of the object:

object of class **UnboundedBuffer**

insert(...)

```
insert into data structure;
```

remove(...)

```
if (empty)
  return failure;
else {
  remove from data
  structure;
}
```

Hidden
implementation with
invariant:

$0 \leq \# \text{ elements}$
stored in data
structure

Several users of the object:

Monitor object of class **UnboundedBuffer**

synchronized
insert(...)

```
insert into data structure;
notify();
```

synchronized
remove(...)

```
if (empty)
  wait();

remove from data
structure;
```

Hidden
implementation with
invariant:

$0 \leq \# \text{ elements}$
stored in data
structure

Second semester (OO) programming course @ University of Oslo, Norway

by Stein Gjessing: Objects with invariants and one vs. several users, version 2

One user of the object:

object of class **BoundedBuffer**

insert(...)

```
if (full)
  return failure;
else {
  insert into data structure;
}
```

remove(...)

```
if (empty)
  return failure;
else {
  remove from data
  structure;
}
```

Hidden
implementation with
invariant:

$0 \leq \# \text{ elements}$
stored in data
structure $\leq \text{MAX}$

Several users of the object:

Monitor object of class **BoundedBuffer**

“synchronized”

insert(...)

```
if (full)
  nonfull.await();

insert into data structure;
nonempty.signal();
```

“synchronized”

remove(...)

```
if (empty)
  nonempty.await();

remove from data
structure;
nonfull.signal();
```

Hidden
implementation with
invariant:

$0 \leq \# \text{ elements}$
stored in data
structure $\leq \text{MAX}$