CMSC 433
Programming Language Technologies and Paradigms

Hoare Logic: Loop Invariants

Hoare Logic Rules: While loops

$$\frac{\{P \;\&\&\; b\}\ s\ \{P\}}{\{P\}\ \texttt{while}\ b\ \{\ s\ \}\ \{P \;\&\&\; !b\}}$$

$P \Rightarrow Inv$

The invariant is initially true

$\{ Inv \;\&\&\; B \}\ S\ \{Inv\}$

Loop preserves the invariant

$(Inv \;\&\&\; !B) \Rightarrow Q$

Invariant and exit implies postcondition

Hoare Logic Rules: While loops

$$\frac{\{P \;\&\&\; b\}\ s\ \{P\}}{\{P\}\ \texttt{while}\ b\ \{\ s\ \}\ \{P \;\&\&\; !b\}}$$

if P is an invariant of s, then no matter how many times the loop body executes, s is going to be true when the loop finally finishes.

P must be strong enough to prove the postcondition and weak enough to be inferred from the precondition.

Practice: Loop Invariants

Consider the following program:

```
{ n >= 0 }
i := 0;
while (i < n){
   i := n;
}
{i = n}
```

Which of the following loop invariants are correct?

A. i = 0

B. i = n

C. n >= 0

D. i <= n

Practice: Loop Invariants

Consider the following program:

```
{ n >= 0 }
i := 0;
while (i < n){
   i := n;
}
{i = n}
```

Which of the following loop invariants are correct?

A. i = 0

B. i = n

C. n >= 0 (irrelevant to the loop)

D. i <= n

Loop Example

```
{ n >= 0}
j := 0;
s := 0;
while (j < n){
  j := j + 1;
  s := s + j;
}
{ s = n * (n+1)/2}  //0+1+2…n
```

Loop Example

```
{ n >= 0}
j := 0;
s := 0;
{ s == j * (j*1)/2}
while (j < n){
  {s == j * (j*1)/2}
  j := j + 1;
  s := s + j;
  {s == j * (j*1)/2}
}
{ s = n * (n+1)/2}  //0+1+2…n
```

Loop Example

```
{ n >= 0}
j := 0;
s := 0;
Assert s == j * (j*1)/2;
while (j < n)
invariant s == j * (j+1)/2
{
  assert s == j * (j*1)/2;
  j := j + 1;
  s := s + j;
  assert s == j * (j*1)/2;
}
{ s = n * (n+1)/2}  //0+1+2…n
```