

**Program**

$p ::= cd_1, \dots, cd_n$  One of which contains a main method declaration

**Class definition**

$cd ::= \text{class } C \text{ extends } C$   
 $\{fd_1 \dots fd_k$   
 $cnd$   
 $md_1 \dots md_n\}$

**Field definition**

$fd ::= Cf;$

**Constructor definition**

$cnd ::= C(C_1x_1, \dots, C_nx_n)\{\text{super}(e_1, \dots, e_k); s_1 \dots s_n\}$

**Method definition**

$md ::= \tau m(C_1x_1, \dots, C_nx_n)\{s_1 \dots s_k\}$

**Return Type**

$\tau ::= C \mid \text{void} \mid \text{int} \mid \text{long} \mid \text{double} \mid \text{boolean} \mid \text{char}$

**Expression**

$e ::= x$	Variable
$null$	Null
$e.f$	Field access
$(C)e$	Cast
$pe$	Promotable expression
$prim$	Primitive

**Primitive**

$prim ::= \text{int} \mid \text{long} \mid \text{double} \mid \text{boolean} \mid \text{char} \mid \text{null}$	Primitive type
$prim \text{ infixop } prim$	Infix op
$(prim)$	Bracketed primitive

**Infix op**

$\text{infixop} ::= + \mid - \mid * \mid / \mid \% \mid > \mid >= \mid \&\& \mid || \mid == \mid !$  infix op

**Promotable expression**

$pe ::= e.m(e_1, \dots, e_k)$	Method invocation
$\text{new } C(e_1, \dots, e_k)$	Object creation

**Statement**

$s ::= ;$	No-op
$pe;$	Promoted expression
$\text{if } (e == e)\{s_1 \dots s_k\} \text{ else } \{s_k + 1 \dots s_n\}$	Conditional
$e.f = e;$	Field assignment
$C x;$	Local variable declaration;
$\text{return } e;$	Return
$\{s_1 \dots s_n\}$	Block