```
type expr =
  | ELet of string * expr * expr
  | EId of string
  | ENum of int


let rec compile_expr e env si = ...
let rec well_formed e env = ...
```

```
interface Expr {

}
class ELet implements Expr {
  String name; Expr val, body;
  ELet(String name, Expr val, Expr body) {
   this.name = name; this.val = val; this.body = body; }

}
class EId implements Expr {
  String name;
  EId(String name) { this.name = name; }

}
class ENum implements Expr {
  int num;
  ENum(int num) { this.num = num; }

}
```

```java
import java.util.ArrayList;
interface ExprV {
  <T> T visit(ExprVisitor<T> v);
}
interface ExprVisitor<T> {
  T visit(ELet e);
  T visit(EId e);
  T visit(ENum e);
  T visit(EPlus e);
}
class ELet implements Expr {
  String name; Expr val, body;
  ELet(String name, Expr val, Expr body) {
    this.name = name; this.val = val; this.body = body;
  }
  public <T> T visit(ExprVisitor<T> v) { return v.visit(this); }
}
class EId implements Expr {
  String name;
  EId(String name) { this.name = name; }
  public <T> T visit(ExprVisitor<T> v) { return v.visit(this); }
}
class ENum implements Expr {
  int num;
  ENum(int num) { this.num = num; }
  public <T> T visit(ExprVisitor<T> v) { return v.visit(this); }
}
class EPlus implements Expr {
  Expr left, right;
  EPlus(Expr left, Expr right) { this.left = left; this.right = right; }
  public <T> T visit(ExprVisitor<T> v) { return v.visit(this); }
}

class CompileVisitor implements ExprVisitor<ArrayList<String>> {

}
class UnboundIdsVisitor implements ExprVisitor<ArrayList<String>> {

}
```