Semantic Analysis

TEACHING ASSISTANT: DAVID TRABISH

Examples

First, let’s try to detect semantic errors…

Examples

```
void main(void) {
   int i = 0;
   int j = 0;
   printf("", i, j);
}
```

Examples

```
void main(void) {
   int i = 0;
   int j = 0;
   printf("", i, j);
}
```

Valid

Examples

```
class A {
public:
   int x;
}
void main(void) {
   A *a;
   a->y;
}
```

Examples

```
class A {
public:
   int x;
}
void main(void) {
   A *a;
   a->y;
}
```

Invalid

Examples

```
class A {
public:
   int x;
}
void main(void) {
   A *a;
   a.x;
}
```

Examples

```
class A {
public:
   int x;
}
void main(void) {
   A *a;
   a.x;
}
```

Invalid

Examples

```
class A {
private:
   int x;
}
void main(void) {
   A *a;
   a->x;
}
```

Examples

```
class A {
private:
   int x;
}
void main(void) {
   A *a;
   a->x;
}
```

Invalid

Examples

```
void main(void) {
  int i = 9 / 0;
}
```

Examples

```
void main(void) {
  int i = 9 / 0;
}
```

Invalid

Examples

```
void main(void) {
   int j = 0;
   int i = 9 / j;
}
```

Examples

```
void main(void) {
   int j = 0;
   int i = 9 / j;
}
```

Valid

Examples

```
void main(void) {
  File *f;
  while (f = 5);
}
```

Examples

```
void main(void) {
   File *f;
   while (f = 5);
}
```

Invalid

Examples

```
void main(void) {
  File *f;
  while (f = (File *)(5));
}
```

Examples

```
void main(void) {
  File *f;
  while (f = (File *)(5));
}
```

Valid

Examples

```
void main(void) {
   7 < “7”;
}
```

Examples

```
void main(void) {
   7 < “7”;
}
```

Invalid

Examples

```
void main(void) {
   “7” < “8”;
}
```

Examples

```
void main(void) {
   “7” < “8”;
}
```

Valid

Examples

```
typedef struct {
   int x;
   int y;
   int z;
   struct point_t next;
} point_t;
```

Examples

```
typedef struct {
   int x;
   int y;
   int z;
   struct point_t next;
} point_t;
```

Invalid

Examples

```
typedef struct {
  int x;
  int y;
  int z;
  struct point_t *next;
} point_t;
```

Examples

```
typedef struct {
   int x;
   int y;
   int z;
   struct point_t *next;
} point_t;
```

Valid

Examples

```
int main(void) {
   return “7”;
}
```

Examples

```
int main(void) {
   return “7”;
}
```

Invalid

Examples

```
int g() {
   return f();
}
int f(void) {
   return 0;
}
```

Examples

```
int g(void) {
  return f();
}
int f(void) {
  return 0;
}
```

Invalid

Examples

Now, let’s check the actual behavior…

Examples

```
void main(void) {
  int i = 1;
  int j = 2;
  int a = 3;
  if ((a = i) || (a = j)) {
    printf("%d\n", a);
  }
}
```

Examples

```
void main(void) {
  int i = 1;
  int j = 2;
  int a = 3;
  if ((a = i) || (a = j)) {
    printf("%d\n", a);
  }
}
```

1

Examples

```
void main(void) {
  int i = 1;
  int j = 2;
  int a = 3;
  if ((a = i) | (a = j)) {
    printf("%d\n", a);
  }
}
```

Examples

```
void main(void) {
  int i = 1;
  int j = 2;
  int a = 3;
  if ((a = i) | (a = j)) {
    printf("%d\n", a);
  }
}
```

2

Examples

```
class Father {
public:
  virtual void print() { printf("100\n"); }
}
class Son : public Father {
  virtual void print() { printf("50\n"); }
}
int main(void) {
  Father *f = new Son();
  f->print();
}
```

Examples

```
class Father {
public:
  virtual void print() { printf("100\n"); }
}
class Son : public Father {
  virtual void print() { printf("50\n"); }
}
int main(void) {
  Father *f = new Son();
  f->print();
}
```

50

Examples

```
class Father {
public:
  virtual void print() { printf("100\n"); }
}
class Son : public Father {
  virtual void print() { printf("50\n"); }
}
int main(void) {
  Son *s = new Father();
  s->print();
}
```

Examples

```
class Father {
public:
  virtual void print() { printf("100\n"); }
}
class Son : public Father {
  virtual void print() { printf("50\n"); }
}
int main(void) {
  Son *s = new Father();
  s->print();
}
```

Examples

```
class Father {
public:
  virtual void print() { printf("100\n"); }
}
class Son : public Father {
  virtual void print() { printf("50\n"); }
  virtual void what() { printf("foo\n"); }
}
void f(Father *f) {
  f->what();
}
int main(void) {
  f(new Son());
}
```

Examples

```
class Father {
public:
  virtual void print() { printf("100\n"); }
}
class Son : public Father {
  virtual void print() { printf("50\n"); }
  virtual void what() { printf("foo\n"); }
}
void f(Father *f) {
  f->what();
}
int main(void) {
  f(new Son());
}
```

Examples

```
void main(void) {
  const int c = 7;
  int *y = (int *)(&c);
  *y = 100;
  printf("%d\n", c);
}
```

Examples

```
void main(void) {
   const int c = 7;
   int *y = (int *)(&c);
   *y = 100;
   printf("%d\n", c);
}
```

100

Examples

```
int main(int argc, char *argv[]) {
  if (argc > 2)
    main(2, argv);
  return 0;
}
```

Examples

```
int main(int argc, char *argv[]) {
  if (argc > 2)
    main(2, argv);
  return 0;
}
```

Valid

Examples

```
void f(char *input) {
  *input = 'A';
}
int main(int argc, char *argv[]) {
  const char *p = "1234";
  f(p);
  return 0;
}
```

Examples

```
void f(char *input) {
  *input = 'A';
}
int main(int argc, char *argv[]) {
  const char *p = "1234";
  f(p);
  return 0;
}
```

Invalid

Examples

```
void main(void) {
  char *p = “aaaa”;
  char *q = “bbbb”;
  if (p > q)
    printf(“1\n”);
  if (p < q)
    printf(“2\n”);
  if (p == q)
    printf(“3\n”);
}
```

Examples

```
void main(void) {
  char *p = “aaaa”;
  char *q = “bbbb”;
  if (p > q)
    printf(“1\n”);
  if (p < q)
    printf(“2\n”);
  if (p == q)
    printf(“3\n”);
}
```

1

2

Examples

```
void main(void) {
  char *p = “aaaa”;
  char *q = “aaaa”;
  if (p > q)
    printf(“1\n”);
  if (p < q)
    printf(“2\n”);
  if (p == q)
    printf(“3\n”);
}
```

Examples

```
void main(void) {
  char *p = “aaaa”;
  char *q = “aaaa”;
  if (p > q)
    printf(“1\n”);
  if (p < q)
    printf(“2\n”);
  if (p == q)
    printf(“3\n”);
}
```

```
3
```