1. There are four violations:

1. The `then` keyword is not used in Scala (static error).
2. Local variable `s` must be initialized as part of its definition (static error).
3. The first branch of the `if` evaluates to `Unit`, which does not match `Int`, the type of the second branch and the declared return type of the function (static error).
4. Since `j = -1`, the reference `a(j)` will be out of bounds (checked runtime error).

Note that Scala has no unchecked runtime errors.

2. (a)

```scala
f : ((pair num a) -> (pair a num))
g : (a -> (list a))
h : ((pair num num) -> (list (pair num num)))
```

(b)

```scala
f : ((pair num num) -> (pair num num))
g : ((pair num num) -> (list (pair num num)))
h : ((pair num num) -> (list (pair num num)))
```

3. The key idea is that since `throw` never yields a value, it can have any type whatever.

```latex
\Gamma \vdash (\text{throw}) : t
\Gamma \vdash e_1 : t \quad \Gamma \vdash e_2 : t
\Gamma \vdash (\text{catch } e_1 \ e_2) : t
```

4. (a)

- OO programmer hacks classes
- Functional programmer uses pattern matching

(b)

- Scala programmer hacks code
- Scala programmer hacks code

5.

```scala
case class P(i: Int, u: T, v: T) extends T {
  def f() = i * u.f() + v.f()
}

case class Q(b: Boolean) extends T {
  def f() = if (b) 1 else 0
}```
6.a. \( f: x, z \mapsto y \).

(b)

\[
\text{def } M2(x: \text{Boolean}, y: \text{Int}, z: \text{Int}) = \\
R(w => \text{if } (x) \text{ z + w else w - 42}, \\
w => w + y)
\]

7. Under method A, \( s \rightarrow {}^{\text{"x"}} \text{ and } {}^{\text{"x"}} \rightarrow s \) will both take time proportional to \(|s|\), because the entire string must be copied. Under method B, \( s \rightarrow {}^{\text{"x"}} \) will still take time proportional to \(|s|\), because the string must be traversed, but \( {}^{\text{"x"}} \rightarrow s \) will take only unit time. So comparing the execution times of the following programs should do the trick: if program 1 runs much more slowly than program 2, method B is being used; if the runtimes are about the same, it's method A.

Program 1:

\[
s = "\)
for i = 1 to 1000000 do \\
s = s ++ "x";
\]

Program 2:

\[
s = "\)
for i = 1 to 1000000 do \\
s = "x" ++ s;
\]

8.

\[
\text{def } \text{count}(b:B, x:A) : \text{Int} = b \text{ match } \\
\text{case } \text{EmptyB} => 0 \\
\text{case } \text{InsertB}(b, y) => \text{count}(b, x) + (\text{if } (x == y) 1 \text{ else } 0) \\
\text{case } \text{DeleteB}(b, y) => (\text{count}(b, x) - (\text{if } (x == y) 1 \text{ else } 0)) \text{ max } 0 \\
\text{case } \text{UnionB}(b1, b2) => \text{count}(b1, x) + \text{count}(b2, x)
\]