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) `int`
   (b) `(a_1 \rightarrow (a_2 \rightarrow a_1))`
   (c) `((\text{bool} \rightarrow a_1) \rightarrow a_1)`
   (d) `((a_1 \rightarrow \text{bool}) \rightarrow (a_1 \rightarrow \text{bool}))`
   (e) `((a_1 \rightarrow (a_2 \rightarrow a_3)) \rightarrow ((a_1 \rightarrow a_2) \rightarrow (a_1 \rightarrow a_3)))`

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

\[
\begin{align*}
TE \vdash (\text{throw}) : t \\
TE \vdash e_1 : t & \quad TE \vdash e_2 : t \\
\overline{TE} \vdash (\text{catch } e_1 e_2) : t
\end{align*}
\]

4. (a)

\begin{itemize}
\item OO programmer hacks classes
\item Functional programmer uses pattern matching
\end{itemize}

(b)

\begin{itemize}
\item Scala programmer hacks code
\item Scala programmer hacks code
\end{itemize}

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. \( f: x.z. g: y. \)

(b)

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

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

Program 1:

\[
s = ""
\text{for } i = 1 \text{ to } 1000000 \text{ do} \\
s = "x" ++ s;
\]

Program 2:

\[
s = ""
\text{for } i = 1 \text{ to } 1000000 \text{ do} \\
s = s ++ "x";
\]

8.

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