1. There are four violations:

1. The \texttt{then} keyword is not used in Scala (static error).
2. Local variable \texttt{s} must be initialized as part of its definition (static error).
3. The first branch of the \texttt{if} evaluates to \texttt{Unit}, which does not match \texttt{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) \texttt{int}
(b) \((a_1 \rightarrow (a_2 \rightarrow a_1))\)
(c)\(((\texttt{bool} \rightarrow a_1) \rightarrow a_1)\)
(d)\(( (a_1 \rightarrow \texttt{bool}) \rightarrow (a_1 \rightarrow \texttt{bool}))\)
(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 \texttt{throw} never yields a value, it can have any type whatever.

\[
\text{TE} \vdash (\text{throw}) : t
\]

\[
\text{TE} \vdash e_1 : t \\
\text{TE} \vdash e_2 : t \\
\text{TE} \vdash (\text{catch } e_1 e_2) : t
\]

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.

\begin{verbatim}
  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
  }
\end{verbatim}

6.a. \( f: x, z, g: y. \)

(b)
def M2(x:Boolean,y:Int,z:Int) = 
  R(w => if (x) z + w 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 = ""
    for i = 1 to 1000000 do
        s = "x" ++ s;

Program 2:

    s = ""
    for i = 1 to 1000000 do
        s = s ++ "x";

8.

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