Define a new exception.

```
exception Name of type
```

What is the type of an exception?

What is the type of `raise`?

```
exn

exn -> 'a
```

Give the `try` syntax.

It's exactly the same as `match`.

Lookup functions that cannot find their
element raise ...

... Not_found.

Functions can reject bad arguments with ...

… Invalid_argument.

What is `Failure`?

How is it different from `Invalid_argument`.

Which is typically handled?

An exception for bad arguments less severe than `Invalid_argument` or not the result of programming errors.

It takes an error string, typically the name of the function doing the raising.

It is only typical to handle `Failure`.

If no pattern is matched, what happens?

A `Match_failure` is raised.

Describe assertions.

```
assert(cond)
```

Raises an `Assert_failure` if the condition fails.

Name the errors resulting from lack of
memory or stack space.

- Out_of_memory
- Stack_overflow

Name some non-typical uses of exceptions.

- Reduce memory usage.
- As a break statement.
- To create *finally* semantics.

How can exceptions reduce memory usage?

Needless reconstruction of the input can be avoided on unsuccessful runs of functions that cons pattern match each element of a list.