Scope of environment variables in shells and shell scripts

A environment variable defined in a shell (or a shell script) stays defined everywhere, including inside new shells (or shell scripts) executed within the shell, until the shell (or the shell script) terminates or the variable gets unset explicitly. E.g.

script1.sh contains:

```
export VAR1=testing
bash script2.sh
```

script2.sh contains:

```
echo $VAR1
```

Then executing script1.sh gives the following:

```
bash-3.00$ bash script1.sh
testing
bash-3.00$ echo $VAR1
bash-3.00$ echo $UNDEFINED_VAR

bash-3.00$
```

If a parent script defines a variable and then a child script redefines it, the child sees the new value but the parent retains the old value. E.g.

script1.sh contains:

```
export VAR1=testing
bash script2.sh
echo $VAR1
```

script2.sh contains:

```
VAR1=overwritten
echo $VAR1
```

Then executing script1.sh gives the following:

```
bash-3.00$ bash script1.sh
overwritten
testing
bash-3.00$
```