CHAPTER 1

Expressions solutions

Here are the possible solutions of the implementation we asked for the Expression Chapter **??**.

1.1 Evaluate message

```
EConstant >> evaluate
  ^ value
```

```
ENegation >> evaluate
  ^ expression evaluate negated
```

```
EAddition >> evaluate
  ^ left evaluate + right evaluate
```

```
EMultiplication  >> evaluate
  ^ left evaluate + right evaluate
```

1.2 Negated message

```
EAddition >> negated
  ^ ENegation new expression: self
```

```
EMultiplication >> negated
  ^ ENegation new expression: self
```

1.3 Better class instance creation interface

```
ENegation class >> expression: anExpression
  ^ self new expression: anExpression
```

```
EMultiplication class >>  left: anExp right: anExp2
  ^ self new left: anExp ; right: anExp2
```

1.4 Printing addition and multiplication

```
EAddition >> printOn: aStream
  aStream nextPutAll: '( '.
  left printOn: aStream.
  aStream nextPutAll: ' + '.
  right printOn: aStream.
  aStream nextPutAll: ' )'
```

```
Emultiplication >> printOn: aStream
  aStream nextPutAll: '( '.
  left printOn: aStream.
  aStream nextPutAll: ' * '.
  right printOn: aStream.
  aStream nextPutAll: ' )'
```

1.5 Negated negation

```
ENegation >> negated
  ^ expression
```

1.6 evaluateWith:

```
EMultiplication >> evaluateWith: anObject
  ^ (right evaluateWith: anObject) + (left evaluateWith: anObject)
```