Converting an LL(1) grammar into Code

X → A

```
void X() {
        A();
}
```

X → AB

```
void X() {
        A();
        B();
}
```

X → xAy

```
void X() {
      Expect(x);
      A();
      Expect(y);
}
```

$X \rightarrow A^*$

```
void X() {
      while (Current == Start(A)) {
            A();
      }
}
```

$X \rightarrow A^{+}$

```
void X() {
      do {
            A();
      } while (Current == Start(A));
}
```

X → A?

```
void X() {
      if (Current == Start(A)) {
            A();
      }
}
```

X → A | B

```
void X() {
      switch (Current) {
      case Start(A): A(); break;
      case Start(B): B(); break;
      default: SyntaxError;
      }
}
```