1. Eine Klasse ist eine Vorlage, nach der ein oder mehrere Objekte instanziiert werden. Objekte bestehen aus der Vorlage, die mit Daten gefüllt werden. Die Objektinstanz ist, wenn ein Objekt hergestellt wird. Eine Klasse hat mehrere Objektinstanzen, die Vorlage bleibt bei jedem gleich, allerdings ändern sich die Daten, die eine Objektinstanz hat.

2. Information hiding ist wichtig, um Daten von einem anderen Teil der Software zu schützen. Information hiding funktioniert, indem man Daten nicht von anderen Teilen zugänglich macht. Dazu benutzt man Kapselung.

3. Klassen kann man in Libraries speichern, die dann einfach zu implementieren sind. Aus einer Klasse kann man mehrere Objekte herstellen. Klassen kann man wie Lego-Steine sehen, da sie sehr modular sind. Man kann Klassen auch von anderen Klassen abrufen, man kann mehrere Unterklassen haben, die durch Vererbungen die Eigenschaften und die Methoden erweitern.

4. Der Wartungsaufwand von Software gibt an, wie viel man testen muss, um sicher zu sein, dass Code richtig funktioniert. Mit Klassen muss man nur wenig testen, da der Code in dem Fall immer gleich arbeitet.

5. Eigenschaften sind die Daten der Klasse, die man beim Objekt verändern kann und benutzen kann, um Methoden auszuführen. Die Daten dieser „Merkmale“ sind bei den Objekten, die aus der gleichen Klasse kommen, verschieden. Sie sind meistens Variablen. Durch public und private kann man die Sichtbarkeit angeben.

6. Methoden sind Funktionen einer Klasse bzw. eines Objektes, die die Daten verarbeiten und benutzen. Methoden kann man z.B. benutzen, um Eigenschaften des Objektes zu lesen oder zu schreiben.

7. Mit public kann man von jedem Teil des Programmes auf der Eigenschaft zugreifen (lesen/schreiben). Mit private kann man nur innerhalb der Klasse auf der Eigenschaft zugreifen (lesen/schreiben). Von außerhalb existiert diese Eigenschaft nicht.

8. Durch get und set -Methoden kann man private-Eigenschaften verändern, das erhöht die Sicherheit der Eigenschaften und der Klasse, da man nicht von einem anderen Teil des Codes auf Eigenschaften fehlerhaft zugreifen kann. Durch get und set- Methoden muss man explizit diese Eigenschaft verändern möchten, was die Möglichkeit einen Fehler zu machen verringert.

9. git init wird benutzt, um eine neue, leere Repository lokal zu initialisieren. Durch git clone wird eine Repository, die schon existiert, lokal geklont (heruntergeladen).

10. git pull ladet die Veränderungen einer remoten Repository lokal herunter. git push ladet die lokalen Veränderungen auf der remoten Repository hoch.

11. Die Arbeitskopie ist der Teil, den man Verändern kann, also die Files die Verändert werden, befinden sich in der Arbeitskopie. Die Repository ist das gesamte Projekt, mit Arbeitskopie, Index und HEAD.

12. Ein Konflikt entsteht, wenn zwei Menschen den gleichen Code Verändern und einen Push machen, man muss dann entscheiden welche Veränderung man behalten will (Konflikt lösen).

13. Git verwendet man, weil es nützlich ist, einen Verlauf der Veränderungen zu haben, um Bugs zu lösen, die erst später eingetreten sind. Git verwendet man, um mit mehr Menschen auf ein Projekt effizienter zu arbeiten. Wenn man einen Dienst wie GitHub verwendet, dient es auch als Backup des Codes.