1. Object- является супер классом от которого происходят все остальные классы. HashCode – генерирует число на основе конкретного объекта (уникальное значение). toString-возвращает значение объекта в виде строки. Equals- сравнивает 2 объекта на равенство. getClass-позволяет получить тип данного объекта. Finalize – деструктор. Clone – создает копию объекта и возвращает на нее ссылку. Notify – снятие паузы. wait – пауза.
2. Хеш-значение – определенный номер ячейки в памяти, в которой хранится объект. И, как сказано выше, хеш-код зависит от реализации объекта, т е если будут разные значения в объекте, то хеш-код будет одинаковый.
3. Для расширения и/или изменения функционала старого класса в новом. Класс человек в него входят имя фамилия возраст и наследуем этот класс, а новый назовем Студент и у него по мимо старых полей и методов появятся ещё и новые который будут описывать более детально этот класс. Поля и методы модификатора private наследуются просто у нас нету к ним доступа.
4. Конструктор базового класса. Super()-конструктор по умолчанию базового класса. Нельзя использовать одновременно super() and This() так как они должны стоять выше всех выражений и операций.
5. Ссылка базового класса может ссылаться на объект своих производных классов но никак не на оборот так как не обладать методами производного типа. Объект производного класса может быть использован вместо объекта базового, но никак не на оборот, что базовый может не иметь того же метода что и производный.
6. Берем метод из базового класса и его видоизменяем в производном. С помощью этого можно изменить 1-модификатор доступа(расширять) 2-возвращаемый тип(иерархии вниз) 3-имена аргумента. Можно переопределять методы родителей или суперклассов.
8. Нельзя изменять поля и переопределять методы модификатора final. Класс final нельзя наследовать и все методы его являются как final. Их используют для повышения защиты и эффективности тестов проверки.
9. Есть явное и неявное преобразование.
10. Class-класс который содержит метаданные о классе(наз пакеты методы поля конструкторы аннотации). Instanceof возвращает значение типа boolean а getClass возвращает объект Class.
12. Equals:
    1. Рефлексивность-для любого заданного значения x, выражение x.equals(x) должно возвращать true. x!=0
    2. Симметричность-для любых заданных значений x и y, x.equals(y) должно возвращать true только в том случае, когда y.equals(x) возвращает true.
    3. Транзитивность-для любых заданных значений x, y и z, если x.equals(y) возвращает true и y.equals(z) возвращает true, x.equals(z) должно вернуть значение true.
    4. Согласованность-для любых заданных значений x и y повторный вызов x.equals(y) будет возвращать значение предыдущего вызова этого метода при условии, что поля, используемые для сравнения этих двух объектов, не изменялись между вызовами.
    5. Сравнение null-для любого заданного значения x вызов x.equals(null) должен возвращать false.

Только в случае если ваш переопределенный метод соответствует всему перечисленному считается что вы выполнили контракт.

HashCode:

1. Вызов метода hashCode один и более раз над одним и тем же объектом должен возвращать одно и то же хэш-значение, при условии, что поля объекта, участвующие в вычислении значения, не изменялись.
2. вызов метода hashCode над двумя объектами должен всегда возвращать одно и то же число, если эти объекты равны (вызов метода equals для этих объектов возвращает true).
3. Вызов метода hashCode над двумя неравными между собой объектами должен возвращать разные хэш-значения. Хотя это требование и не является обязательным, следует учитывать, что его выполнение положительно повлияет на производительность работы хэш-таблиц.

toString:

1. Должен вызываться для всех полей объекта.
2. Всегда возвращает String