Q. What is method overloading? explain with e.g

=> Concept of calling / passing messages to ~~same~~ method having same name is method overloading

e.g. sum(int num1);
sum (int num2, float num3);

Above 2 method has same name but diff. parameter in quantity.

Q. method overloading resolutⁿ? how java determine which method to call.

->
1) by passing different number of ~~parameter~~ argument
2) by passing different data type as ~~parameter~~ argument
3) by changing the order of ~~parameter~~ argument

java looks at argument & determines which method to be called by comparing arg. with par.

Q. Static keyword? diff. betⁿ static & non-static.

=> It means the member of class are called / initialized only once for that particular class.

The main diff is we can create object & instance of a not static field & method corresponding to that instance is a non-static method. ; ~~can be created multiple times here.~~
~~equal~~ to

static methods are those which does not need instance for initialization & called once per class

Q. Can static method be overloaded
how static variable are shared across multiple instance of class.

→. yes,
when we create object instance; field of instance (non-static) gets space inside instance but static field gets space once per class. & when called for printing we don't need to call static variable explicitely inside non-static method / constructor.

Q. static keyword in context of memory man.

→. As static field / method gets stored & initialised at time of class loading they get space only once while non-static gets space on every instance call. thus reducing memory utilizat^{n}. (optimal mem. utilizat^{n} is achieved).

Q. final keyword.

→. used for making constant wheather it's field, object reference or local method variable, once being initialized cannot be assigned / reassigned / modified.

Q. this keyword., use of this in constructor & meth

→. An implicit ref provided by compiler which points towards the field of class.

this keyword can only be used in same class of the field which it needs to access.

use in const. & method.

↳ Consider field.

↳ int num;,

we will use this.num = num; in constructor as well as in method when name of parameter of ctor & method is same

~~if we use print statement in same class~~

Q. narrowing & widening

=>. Converting lower data type (primitive) to higher & higher to lower is termed as widening & narrowing resp.

Q. E.g of narrowing & widening

→ wide { int num1 = 10;
double num2 = num1;.

narrow { double num4 = 27.55;
byte num0 = (byte) num4;.

use of type casting.

Q. How java handle loss in narrowing?

→ we need to explicitly provide type casting method, if not provided it throws error.

Q. Automatic widening.

→ In widening we need to specify the type casting explicitely compiler does that implicitely as there is no loss of data.

Q. Implicit of widening & narrowing

→ widening ⇒ As higher data type conversion is happening the size of the resp. data type is considered & more mem is utilized, & for float & double decimal values comes in picture.

Narrowing ⇒ If we consider float & double the decimal values are lost & data loss

another way data loss is when we narrow down from higher to lower say from int to byte, & then widen the same o/p from byte to int there is chance of data loss.

e.g. int num = 300;

↳ narrow to byte ⇒ range is -128 to 127.

↳ after narrow byte will store 44.

when we widen again to int.

it will be 44, instead of 300.