Allignment - 6

① method overloading

→ to achieve Polymorphism, we do method overloading
→ method has same name,
diff Parameters ~~arg~~, diff seq, diff types,
via it we Can achieve it

e.g.

```
Class Program
{       int a=5;

    void print(int b) // method overloading
      {
         a=b;  System.out.println(a); //10
      }

    void print() // method overloading
      { System.out.println(a); // 5
      }
}

class Test
{ public static void main(String[] args)
  {
    Program p=new Program();
        p.print(10);
        p.print();
  }
}
```

M T W T F S S

Page No.
Date
YOUVA

(2) Rules for method overloading resolution
→
(1) ~~[illegible]~~ has same name
(2) diff. types, diff seq parameters diff no. args

(3)

How does java determine which overloaded method to call?
→
→ it gets resolve at compile time,
→ compile binds call, to definition of method depends on types, args, seq.

here in previous e.g.
p.print(10);
p.print();
here print has one ~~[illegible]~~ parameter & another has no ~~[illegible]~~ parameters,

→ compiler understand whom to bind, by seeing definition of method.

(3) static keyword.
→ → It's associate with class
→ can be methods, variables, static inner classes
→ when we define something as static,
it gets memory at class loading time in method area
→ all objects share's memory of static data

static method	Non-static method
- no need to make obj to call it	- via obj we have to call it
- class level method	- instance method
- declared with static	- declare without static
- used to access data with class.name	- used to access data with obj

Scanned with OKEN Scanner

Page No.:
Date:

④ Can static methods overloaded, overridden in java

→

→ we can overload static methods
cuz it's in same class,

→ we can't override static methods, it's associated
with class, so can't override in another class.

How static variables shared across multiple instances
of class

→ static variable → class level variable,
stored in method ~~[illegible]~~ area,
& gets shared in every instance

⑤ what is role of static keyword in respect to memory
mgmt

→ if something, data is common in every obj
make it static, cuz only one memory get, no
need to create again n again in every obj,
→ all obj will share it from method area.
→ it saves memory.

⑥ significance of final keyword

→

→ if something is constant, make it final
→ if once you give value to it
by initilizaon OR
by assignment
then u can't reassign it, it will
give compilation error.
→ we have to use it to store constant data.
→ if want no one will use our class, make it final
→ can use with variable, methods, class.

7) Can a final method be overridden in subclass, How does the final keyword affect variables, methods, & classes in java.

→

→ u can't override.
→ if var is final, u can't change value
→ if method is final, u can't override
→ if classes is final, u can't inherit

if u want, your implementation should have to be same, use ~~[illegible]~~ final, so no one can modify it

8) what does this keyword represent in java. How is this keyword used in class & method

→ this is reference variable points to current instant / obj

```
→ class Program
{ int a, b;
  public Program (int a, int b)
  {
     this.a = a;
     this.b = b;
  }
  public void print()
  {
     s.o.p( this.a + " " + this.b");
  }
}
```

```
class Test
{
  p s v m (String[] args)
  {
     Program p = new ~~Progr~~
        Program(10,20);
     p.print();
  }
}
```

→ when methods calls on ref, this also goes in method, same, as instant gets create, this gets created, then it gets passed in constructor.

9. what are narrowing, widening conversion

→

narrowing :
- big range data, conversion to small range data, we do it forcefully
- data loss happens

```
int a = 2;
double b = 3.5;
   int c = (int) b;  // narrowing
```

widening :
- small range data gets convert in large range data
- no data loss
- it happens automatically

```
int a = 3
   double d = a;  // widening.
```

10. example of narrowing, widening in primitive data type

→ narrowing :

```
          double data = 5.8;
          int n = 0;
               n = (int) data;
```

widening :

```
          int data = 2;
          double b = 0;
               b = data;
```

M T W T F S S
Page No.: YOUVA
Date:

(11) How does Java Handle potential loss of precision during narrowing conversions

→ you automatically can't do it
→ Java gives Compilation error
→ So we have convert it in respected data type, then only we can do it.

(12) concept of automatic widening

→ widening :-
Conversion of small range data into big data.
- no data loss happens
- big data type, can store value of small data type,
so it happens automatically by Compiler

(13) what & implications of narrowing & widening. Conversion

→ narrowing : data loss happens
widening : no loss happens