Q.Class: Handling multiple class.

Class is a collection of objects, it is also called as blueprint/ logical entity. Java is a true objectoriented language and therefore the underline structure of all java program is classes. Anything we wished to represent in a Java program must be encapsulated in a class that defines the state and behaviour of the basic program components known as object.

Class:

- Pre-defined
  - Scanner
  - Console
  - System
  - String
- User-defined

A class which is created by java programmer is called user-defined class.

For example:

```
Class class_name{
------ // data
------ // methods
}
```

T. Method overloading

Whenever a class contain more than one method with same and different types of parameters called method overloading.

Syntax: return_type method_name(para1);

return_type method_name(para1,para2);

Let take an example:

```
class A
{
void add() {
int a=10, b=20, c;
c=a+b;
System.out.println(c);
}
void add(int x, int y) {
int c;
c=x+y;
System.out.println(c);
}
void add(int x, double y) {
double c;
c=x+y;
System.out.println(c);
}
}
public class overload {
public static void main(String[] args) {
```

```
A r= new A();
r.add();
r.add(20, 34);
r.add(32, 3.456);
}
}
```

• Runtime Polymorphism

A polymorphism which exist at the time of execution of program is called runtime polymorphism.

***Method overriding**

Whenever we writing method in super and sub classes in such a way that method name and parameter must be same called method overriding.

Let us take an example:

```
class shape{
void draw()
{
System.out.println("cant say shape type");
}
}
class square extends shape
{
void draw()
{
System.out.println("square shape");
}
}
class demo
{
public static void main(String[] args) {
shape r=new square();
r.draw();
}
}
```

Q. Difference between Overloading and Overriding.

Overloading	Overriding
1) Whenever a class contain more than one method with same and different types of parameters called method overloading.	1) Whenever we writing method in super and sub classes in such a way that method name and parameter must be same called method overriding.
2) This is relationship between methods in a same class.	2) This is relationship between superclass method and subclass method.
3) May have different return data type.	3) Must have same return type.
4) This is an example of compile time polymorphism.	4) This is an example of run time polymorphism.

Q. *How to define constructor in java. Give example.

Constructor is a special type of method whose name is same as class name.

1. The main purpose of constructor is initialize the object.
2. Every Java class has constructor.
3. A constructor is automatically called at the time of object creation.
4. A constructor never contain any return type including void.

Let us take an example:

```
class A{
int a; String name;
A()
{
a=0; name=null;
}
void show()
{
System.out.println(a+" "+name);
}
}
class B{
public static void main(String[] args) {
A ref=new A();
ref.show();
}
}
```

Q. What do you mean by constructor overloading.

Whenever we have more than one constructor in our class then it is called constructor overloading.

Let us take an example:

```
class con
{
int a; double b; String c;
con()
{
a=100; b=45.45; c="masuk";
}
con(int x)
{
a=x;
}
con(double y, String z)
{
b=y; c=z;
}
}
public class Construc
{
public static void main(String[] args) {
con r=new con();
con r2=new con(10);
con r3=new con(23.56, "mass");
System.out.println(r.a+" "+r.b+" "+r.c);
System.out.println(r2.a);
System.out.println(r3.b+" "+r3.c);
}
}
```