OOps in java is to improve code readability and reusability by defining a Java program efficiently. The main principles of object-oriented programming are abstraction, encapsulation, inheritance, and polymorphism. These concepts aim to implement real-world entities in programs.

Following are the main concepts of oops :

1) Objects
2) Classes
3) Object
4) Class
5) Abstraction
6) Inheritance
7) Polymorphism
8) Encapsulation

objects are always called instances of a class which are created from a class in java

```
Public class Test {
int x=10;
Public static void main (String args []) {
Test t= new Test ();
System.out.println(t.x);
}
}
```

Abstraction is a process which displays only the information needed and hides the unnecessary information. the main purpose of abstraction is data hiding. Abstraction means selecting data from a large number of data to show the information needed, which helps in reducing programming complexity and efforts.

```
//abstract parent class
Abstract class animal {
  //abstract method
public abstract void sound ( ) ;
}
 Public class lion extends animal {
Public void sound ( ) {
        System.out.println (“ roar “ );
        }
public Static void main ( String args [ ] ) {
 animal obj = new lion ( );
obj. sound ();
}
}
```

The output of the program is roar.

Inheritance is the object-oriented programming concept where an object is based on another object. Inheritance is the mechanism of code reuse. The object that is getting inherited is called the superclass and the object that inherits the superclass is called a subclass. We use extends keyword in java to implement inheritance. Below is a simple example of inheritance in java.

```
//create a superclass
Class Add {
int my;
int by;
void setmyby (int xy, int hy) {
my=xy;
by=hy;
}
}
/create a sub class
class b extends add {
int total;
void sum () {
public Static void main (String args [ ] ) {
b subOb= new b ( );
subOb. Setmyby (10, 12);
subOb. Sum ( ) ;
System.out.println(“total =” + subOb. Total);
}
}
```

1) A
2) A
3) B
4) A
5) A
6) C
7) A
8) (B) the output of the program is “0”. As the Variable ‘i’ has not been initialized so, by default the value of the variable is zero.
9) A
10) The output of the program is: - Derived:: show () call

    Explanation: -

    In this program we have two classed one is the superclass named “Base”. Inside the Superclass the method named ”show()” is used to print the statement “Base::show() called”.

    Inheritance is used to inherit the methods and attributes from the “base class” here Class “Derived “is the subclass which inherits the superclass which is “base class” which is done using the keyword “extends”. In the subclass there is a method named “ named ”show()” which is used to print the statement “ Derived::show() called”.

    In the Main method an object of the class derived is created using base class and the method show() is called using the object created named “b”.

    “b.show()” will call the method In the derived class which is public Void show() and this method will execute and will print the statement “Derived::show() called”.

11) The output of the program is: - Compile time error.

    Explanation: -

    In this program The superclass named “ base“ has a method which is defined as “final ” with the method name “show()”.

    Class “derived ” which inherits the superclass “base” using the keyword “extends” has a method show() which is used to print a statement “Derived::show() called ”.

    In the main method class Main is created and inside main method object of Derived class is created using variable “b” . the method “show()” is called using the object created.

    The program will result in a “compile error “ because the derived class cannot override the method from the superclass “final public void show()” because the method has been declared as final and in Java the final keyword is a non-access modifier used for classes and attributes and methods in order to make them non-changeable and make them impossible to inherit or override by any other class.

12) The output of the program is :- Derived::show() called

Explanation :- In this program inside the superclass “base” a method has been created named show() the method has been declared as public static and it is used to print the statement “Base::show() called”.

The subclass “Derived ” inherits the superclass using keyword “extends”. This method will print the statement “Derived::show() called”.

In the main method an object of derived class is created and the method “show()” is called using the object created. When the method show() is called it will run the method in the derived class and print the statement “ Derived::show() called”.

In the program there is a warning which is shown in while calling the method using the object created. The reason behind this is that the methods which are declared as static should be called using the class name rather than creating and object and calling the method. In order to call the static methods there is no need of creating object.

13) The output of the program is : - compile error.

Explanation : - In this program we will get a compile time error as there is no main method is found in the public class file. In order to run the program the main method needs to be defined in the public class file.

14) The output of the program is : - Compile error.

Explanation : - In this program, A superclass named “Derived ” has been created and inside the Superclass A method “ void getDetails() ” has been created which takes string as an input which is stored inside a variable “temp”. This method will print the statement “Deriver class” and the Input given through the Variable “temp” (“Derived class” + temp).

A public class has been created (class Test) which inherits the superclass “class Derived” using the keyword “extends”. Inside the class a method “int getDetails()” is created which has return type int and this method will take input as String and store the value inside the variable temp. this will give us a compile error because the return type of the method is int and the class from which it inherits the methods and parameters are of string type. So, the program will give a compile error.

Also, the main method must be defined inside the public class. In this program, the main method has not been defined inside the public class so, the program will not execute.

15) The output of the program is :-

Adding to 100, x = 104

Adding to 0, y = 3 3 3

Explanation: - In this program, we have a class “test”. Inside class test a variable(“y”) has been defined which has been defined public static and the variable has been initialized to 0.

A class “HasStatic” has been defined, inside the class a Variable(“x”) has been defined as private and static and x has been initialized to 100. In the main method first an object of class HasStatic is created (“hs1”), now the value of “x” is incremented using the object created as(“hs1.x++”). Here, the value of x becomes 101. A new object is created with name hs2 and the value of x is incremented (hs2.x++) and the value of x is 102. New object of HasStatic is created and the value of x is incremented by hs1.x++, now the value of x becomes 103. Now the value of x is Increment by HasStaic,x++. Now, the value of x becomes 104. The output is printed as “Adding to 100, x= 104.

New object is created as t1 and the value of y is increment using t1.y++, the value of y becomes 1. New object t2 is created and the value of y is incremented using t2.y++. The value of y becomes 2. New object of t1 is created and the value of y is increment as t1.y++, now the value of y becomes 3. The print statement will print the output Adding to 0, y = 3 3 3

16) The output of the program is :- compile error.

Explanation : - The output of the program will result in compile error as the The method m1(int i, float f) is ambiguous for the type San because both int and float class have Object as parent class and there is no inheritance. So java compiler doesn’t consider any of them to be more specific, hence the method ambiguous call error.

17) The output of the program is : - compile error.

Explanation : - The output of the program will result in a compile error. In the main method for temp variable of type int the value which has been assigned is null and while compiling the program it will give error as null value cannot be converted into int type.

18) The output of the program is : - 0 0

Explanation : - In this program, a class Test is created and inside the class two variables (x and y) of int type is defined and has been defined as protected.

In the main method an object of class Test is created and the print statement will print the output as (0 0).

19) The output of the program is : - Constructor called 10

Constructor called 5

Explanation :- In this program a class Test1 is created and inside the class constructor Test1 is created which takes int x as parameter. And it will print the statement (constructor called “ + x). another class Test2 is created and inside the class object of the class Test1 is created and value of X is passed as 10. In the same class a constructor Test2 is created which takes int as parameter and the object of Test1 is created and the value is passed through i. in the main method object of Test2 class is created and the value is passed as 5. When the main method is called first it will call the Test1 method and print the statement “Constructor called 10” and it will also run the Test 2 method and print the statement “Constructor called 5”.

20) The output of the program is 7.

21) The output of the program is 2.

Explanation : - In this program, a superclass (class A) is created inside the class a variable int i is created and a method display() is created which will print the value of i.

An subclass (Class B) is created which inherits the superclass “class A”. inside the class a variable j is defined and method display() is created which will print the value of j.

In the main method, object of Class B is created (obj2). The value of i is set to 1 and the value of j is set to 2. The value of obj2 is assigned to r and the method is display() is called. It will call the display method in class B and it will print the value of j which is 2.

22) The output of the program is 2.

Explanation : - In this program, superclass A has been defined and inside the class a variable i is defined which is of int type. Inside the class method display() is created which will print the value of i. A subclass B is created which inherits the superclass A. inside the subclass a variable j is defined which is of type int. inside the class a method display() is created which will print the value of j.

In the main method an object of class B is created and the value of i and j is assigned using the instance of the object. When the method obj.display () is called it will print the value of j which is 2.

23) The output of the program is : - 1 2

Explanation : - In this program inside super class A has two variables defined as public int i and protected int j. A subclass B is created which inherits the superclass A . inside the subclass variable int j is defined. Inside the subclass a method display() is created and inside the method super keyword is used to assign the value for j to 3 and print the value of j.

In the main method an object of class B is created and the value of i and j is assigned using the object instance. Method display is called and it will print the value of i and j (1 2).

24) The output of the program is : - 0 0

Explanation : - In this program, the superclass A is created and inside the class variables i and j has been defined as public. Inside the class constructor A() is created inside the constructor the values of i and j has been defined as 1 and 2. In the subclass B , variable int a has been defined and a constructor B() has been created inside the constructor super() is used to call the parent method.

In the main method an object of class B is created and print the values of i and j. since the object is created and the method is not called the output of the program is 0 0.

25) The output of the program is : -

obj1.a = 4 obj1.b = 3

obj2.a = 4 obj1.b = 3