1) Object-oriented programming languages provide reusability feature using\_\_\_\_\_\_\_

a) Polymorphism

b) Parameters

**c) Inheritance**

d) All of these

2) \_\_\_\_\_\_allows us to build new class with added capabilities by extending existing class.

a) Polymorphism

b) Parameters

**c) Inheritance**

d) All of these

3) Inheritance models\_\_\_\_\_\_relationship between two classes.

**a) 'is-a'**

b) 'its-a'

c) 'has-a'

d) 'is-at'

4) \_\_\_\_\_class is also referred to as super class or base class.

a) Child

**b) Parent**

c) Sub

d) None of these

5) In the same way; child class is also referred to as\_\_\_\_\_derived class or extended class.

a) Inherited Class

b) Parent Class

**c) Sub Class**

d) Super class

6) A\_\_\_\_\_\_\_\_inherits all instance variables and methods from super class and it may have its own added variables and methods.

a) Inherited Class

b) Parent Class

**c) Sub Class**

d) Super class

7) Constructors are not inherited in\_\_\_\_\_\_\_

a) Inherited Class

b) Parent Class

**c) Sub Class**

d) Super class

8) \_\_\_\_\_\_inherits all methods of super class and it may have additional methods.

a) Inherited Class

b) Parent Class

**c) Sub Class**

d) Super class

9) A sub class is not a subset of\_\_\_\_\_\_In fact, sub class usually contains more information and methods than its super class.

a) Inherited Class

b) Parent Class

c) Sub Class

**d) Super class**

10) When an object of\_\_\_\_\_is instantiated, memory is allocated for all its attributes including inherited ones.

a) Inherited Class

b) Parent Class

**c) Sub Class**

d) Super class

11) In Java, to create a sub class, keyword '\_\_\_\_\_' is used in the class definition.

**a) Extends**

b) Int

c) Float

d) Create

12) If the visibility of instance variables of\_\_\_\_\_\_\_is changed to private, the variables are not directly accessible outside the class.

a) Inherited Class

b) Parent Class

c) Sub Class

**d) Super class**

13) \_\_\_\_\_\_\_members are directly available only in the class in which they are defined and nowhere else.

a) Public

**b) Private**

c) Protected

d) Packaged

14) \_\_\_\_\_\_\_members are available as 'private' members in the inherited sub class.

a) Public

b) Private

**c) Protected**

15)Java does not support multiple\_\_\_\_\_\_\_

a) Polymorphism

b) Parameters

**c) Inheritance**

d) All of these

16) A sub class can be derived from only one\_\_\_\_\_\_\_

a) Inherited Class

b) Parent Class

**c) Sub Class**

d) Super class

17) Which of the following defines attributes and methods ?

**a) Class**

b) Object

c) Instance

d) Variable

18) Which of the following keyword is used to declare Class variables and class methods ?

**a) Static**

b) Private

c) Public

d) Package

19) Java does not support multiple\_\_\_\_\_\_\_\_\_\_

a) Polymorphism

b) Parameters

**c) Inheritance**

d) All of these

20) Inheritance means

a) Sub class extends Base class

**b) Sub class extends super class**

c) Sub class create object of super class

d) All of the above

21) What type of inheritance does Java have?

**a) Single Inheritance**

b) Double Inheritance

c) Multiple Inheritance

d) Class Inheritance

22) Which of the following is not an advantage to using inheritance?

a) Similar classes can be made to behave consistently.

**b) One big superclass can be used instead of many little classes.**

c) ode that is shared between classes needs to be written only once.

d) Enhancements to a base class will automatically be applied to derived classes.

23) What is not type of inheritance?

a) Single inheritance

**b) Double inheritance**

c) Multiple inheritance

d) Hierarchical inheritance

24) What are the features reused using Inheritance in Java?

a) Variables

b) Constants

c) Methods

**d) All the above**

24) What is the maximum number of levels possible in a Multilevel Inheritance in Java?

a) 8

b) 16

c) 32

**d) No maximum level**

25) Which inheritance in java programming is not supported

a) Single inheritance

b) Multilevel inheritance

**c) Multiple inheritance using classes**

d) Multiple inheritance using interfaces

26) Which of these keywords is used to refer to member of base class from a sub class?

**a) super**

b) upper

c) this

d) None of the mentioned

27) Which of these is correct way of inheriting class A by class B?

**a) class B extends A {}**

b) class B + class A {}

c) class B extends class A {}

d) class B inherits class A {}

28) Using which of the following, multiple inheritance in Java can be implemented?

**A. Interfaces**

B. Multithreading

C. Private methods

D. Protected methods

29) All classes in Java are inherited from which class?

a) java.lang.class

b) java.class.object

**c) java.lang.Object**

d) java.class.inherited

30) What will be the output of the following Java program?

class A

{

int i;

void display()

{

System.out.println(i);

}

}

class B extends A

{

int j;

void display()

{

System.out.println(j);

}

}

class inheritance\_demo

{

public static void main(String args[])

{

B obj = new B();

obj.i=1;

obj.j=2;

obj.display();

}

}

a) 0

b) 1

**c) 2**

d) Compilation Error

31) What will be the output of the following Java program?

class A

{

int i;

}

class B extends A

{

int j;

void display()

{

super.i = j + 1;

System.out.println(j + " " + i);

}

}

class inheritance

{

public static void main(String args[])

{

B obj = new B();

obj.i=1;

obj.j=2;

obj.display();

}

}

a) 2 2

b) 3 3

**c) 2 3**

d) 3 2

32) What would be the result if a class extends two interfaces and both have a method with same name and signature? Lets assume that the class is not implementing that method.

a) Runtime error

**b) Compile time error**

c) Code runs successfully

d) First called method is executed successfully

33) What is the result of compiling and running the following code?

class Base{

public Base(){

System.out.print("Base");

}

}

class HelloWorld extends Base {

public HelloWorld(){

this("Examveda");

System.out.print("Derived");

}

public HelloWorld(String s){

System.out.print(s);

}

public static void main(String[] args){

new HelloWorld();

}

}

a) ExamvedaDerived

b) ExamvedaBaseDerived

**c) BaseExamvedaDerived**

d) ExamvedaDerivedBase