1)Which of these class is superclass of every class in Java?

a) String class

**b) Object class**

c) Abstract class

d) ArrayList class

2) Which of these method of Object class can clone an object?

a) Objectcopy()

b) copy()

**c) Object clone()**

d) clone()

3) Which of these method of Object class is used to obtain class of an object at run time?

a) get()

b) void getclass()

**c) Class getclass()**

d) None of the mentioned

4) What will be the output of the following Java code?

class Output

{

public static void main(String args[])

{

Object obj = new Object();

System.out.print(obj.getclass());

}

}

a) Object

b) class Object

**c) class java.lang.Object**

d) Compilation Error

5) What is an object in Java?

a) static reference

b) template or blueprint

**c) instance of class**

d) None of above

6) How many characteristics Object has?

a) 2

**b) 3**

c) 5

d) 1

7) Which are the object characteristics?

a) State

b) Behavior

c) Identity

**d) All of above**

8) Which is the best definition of an object?

a) An object is a real-world entity

b) An object is a runtime entity

c) The object is an instance of a class

**d) All of above**

9) What is a class in Java?

a) static reference

**b) template or blueprint**

c) instance of class

d) None of above

10) A Java class can have

a) Fields and Methods

b) Constructors and Blocks

c) Nested class and interface

**d) All of above**

11) Variable of class type are also referred to as\_\_\_\_\_\_\_\_variables.

a) Orientation

b) Position

**c) Reference**

d) Indication

12) Using\_\_\_\_\_\_\_\_keyword, an object can be created.

a) Int

b) Float

**c) New**

d) Real

13) Operator\_\_\_\_\_\_\_\_allocates the memory for an object and returns the address of the object for later use.

a) Int

b) Float

**c) New**

d) Real

14) \_\_\_\_\_\_\_\_\_is the address of the memory location where the object is stored.

a) Memory

b) Variable

**c) Reference**

d) None of these

15) There is a special portion of memory called the\_\_\_\_\_\_where the objects live.

**a) Heap**

b) Pile

c) Stack

d) All of these

16) When an object is created, in addition to allocating memory, a special method called\_\_\_\_\_\_\_is executed to perform initial task.

a) Function

**b) Constructor**

c) Class

d) Method

17) An object can be created of type Room and assign its address to variable rl as\_\_\_\_\_\_\_

**a) rl = new Room();**

b) rl = Room() new;

c) rl = Class Room();

d) None of these

18) With empty parentheses without arguments, a default\_\_\_\_\_\_is called.

a) Function

**b) Constructor**

c) Class

d) Method

19) A\_\_\_\_\_\_initializes the attributes (variables) of the object using default values.

a) Function

**b) Constructor**

c) Class

d) Method

20) The\_\_\_\_\_\_can contain arguments that determine the initial values of variables.

a) Room

b) Brackets

**c) Parentheses**

d) Class

21) Observe the following Room r2 = new RoomO : - Variable r2 contains a\_\_\_\_\_\_or address of memory location where a new object is created.

a) Memory

b) Variable

**c) Reference**

d) None of these

22) The class determines only the\_\_\_\_\_\_\_of the variables.

**a) Types**

b) Collection

c) Location

d) Set

23) The actual\_\_\_\_\_\_\_\_is contained inside the individual objects and not in the class.

a) Information

**b) Data**

c) Collection

d) Variables

24) Every\_\_\_\_\_\_has its own set of data.

a) Class

b) Variable

c) Operator

**d) Object**

25) \_\_\_\_\_\_\_allocated different memory space to hold their data values.

a) Classes

b) Variables

c) Operators

**d) Objects**

26) In Java, when\_\_\_\_\_\_are no more needed, the memory is claimed back for reuse.

a) Classes

b) Variables

c) Operators

**d) Objects**

27) Java has a garbage collector that looks for unused\_\_\_\_\_\_and reclaims the memory that those objects are using.

**a) Objects**

b) Variables

c) Cells

d) Memory spaces

28) In Java, there is no requirement to do any explicit freeing of\_\_\_\_\_\_

a) Cells

b) Variables

**c) Memory**

d) Class

29) In Object-oriented programming (OOP) languages, creating an object is also called\_\_\_\_\_\_instantiation.

a) Class

**b) Object**

c) Inheritance

d) Polymorphism

30) \_\_\_\_\_for an object is created by allocating memory to store data for that object.

**a) Instance**

b) Example

c) Illustration

d) None of these

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

a) Polymorphism

b) Parameters

**c) Inheritance**

d) All of these

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

a) Polymorphism

b) Parameters

**c) Inheritance**

d) All of these

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

**a) 'is-a'**

b) 'its-a'

c) 'has-a'

d) 'is-at'

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

a) Child

**b) Parent**

c) Sub

d) None of these

35) 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

36) 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

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

a) Inherited Class

b) Parent Class

**c) Sub Class**

d) Super class

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

a) Inherited Class

b) Parent Class

**c) Sub Class**

d) Super class

39) 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**

40) 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

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

**a) Extends**

b) Int

c) Float

d) Create

42) 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**

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

a) Public

**b) Private**

c) Protected

d) Packaged

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

a) Public

b) Private

**c) Protected**

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

a) Polymorphism

b) Parameters

**c) Inheritance**

d) All of these

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

a) Inherited Class

b) Parent Class

**c) Sub Class**

d) Super class

47) Which of the following defines attributes and methods ?

**a) Class**

b) Object

c) Instance

d) Variable

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

**a) Static**

b) Private

c) Public

d) Package

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

a) Polymorphism

b) Parameters

**c) Inheritance**

d) All of these

50) 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

51) What type of inheritance does Java have?

**a) Single Inheritance**

b) Double Inheritance

c) Multiple Inheritance

d) Class Inheritance

52) 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.

53) What is not type of inheritance?

a) Single inheritance

**b) Double inheritance**

c) Multiple inheritance

d) Hierarchical inheritance

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

a) Variables

b) Constants

c) Methods

**d) All the above**

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

a) 8

b) 16

c) 32

**d) No maximum level**

56) Which inheritance in java programming is not supported

a) Single inheritance

b) Multilevel inheritance

**c) Multiple inheritance using classes**

d) Multiple inheritance using interfaces

57) 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

58) 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 {}

59) 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

60) 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

61) 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

62) 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

63) 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

64) Which of the following class definitions defines a legal abstract class?

a) class A { abstract void unfinished() { } }

b) class A { abstract void unfinished(); }

**c) abstract class A { abstract void unfinished(); }**

d) public class abstract A { abstract void unfinished(); }

65) Which of the following declares an abstract method in an abstract Java class?

a) public abstract method();

**b) public abstract void method();**

c) public void method() {}

d) public void abstract Method()

66) Which of the following statements regarding abstract classes are true?

a) An abstract class can be extended.

b) A subclass of a non-abstract superclass can be abstract.

c) A subclass can override a concrete method in a superclass to declare it abstract.

**d) All of the above**

67) Determine output of the following code.

interface A { }

class C { }

class D extends C { }

class B extends D implements A { }

public class Test extends Thread{

public static void main(String[] args){

B b = new B();

if (b instanceof A)

System.out.println("b is an instance of A");

if (b instanceof C)

System.out.println("b is an instance of C");

}

}

a) Nothing.

b) b is an instance of A.

c) b is an instance of C.

**d) b is an instance of A followed by b is an instance of C.**

68)In Java, declaring a class abstract is useful

a) To prevent developers from further extending the class.

**b) When it doesn't make sense to have objects of that class.**

c) When default implementations of some methods are not desirable.

d) To force developers to extend the class not to use its capabilities.

69) What is the output for the below code ?

interface A{

public void printValue();

}

public class Test{

public static void main (String[] args){

A a1 = new A(){

public void printValue(){

System.out.println("A");

}

};

a1.printValue();

}

}

a) Compilation fails due to an error on line 3

**b) A**

c) Compilation fails due to an error on line 8

d) null

70) What will be the output?

public interface InfA{

protected String getName();

}

public class Test implements InfA{

public String getName(){

return "test-name";

}

public static void main (String[] args){

Test t = new Test();

System.out.println(t.getName());

}

}

a) test-name

b) Compilation fails due to an error on lines 1

**c) Compilation fails due to an error on lines 2**

d) Compilation succeed but Runtime Exception

71) What will be the output for the below code ?

public interface TestInf{

int i =10;

}

public class Test{

public static void main(String... args){

TestInf.i=12;

System.out.println(TestInf.i);

}

}

**a) Compile with error**

b) 10

c) 12

d) Runtime Exception

72) A class which is declared with the \_\_\_\_\_\_\_\_ keyword is known as an abstract class in Java.

**a) abstract**

b) util

c) extends

d) None of the above

73) Abstract class can have constructors and static methods?

**a) TRUE**

b) FALSE

c) Abstract class can have constructors but can not have static methods.

d) Abstract class can not have constructors but can have static methods.

74)What is the syntax of abstract class in java?

a) abstract A{}

b) abstract class A

**c) abstract class A{}**

d) abstract class A[]

75) Which of these packages contains abstract keyword?

**a) java.lang**

b) java.util

c) java.io

d) java.system

76)An abstract class can have a data member, abstract method, method body (non-abstract method), constructor, and even main() method.

**a) TRUE**

b) FALSE

c) Can be true or false

d) can not say

77)Which of these is not a correct statement?

a) Every class containing abstract method must be declared abstract

b) Abstract class defines only the structure of the class not its implementation

**c) Abstract class can be initiated by new operator**

d) Abstract class can be inherited

78)If a class inheriting an abstract class does not define all of its function then it will be known as?

**a) Abstract**

b) A simple class

c) Static class

d) None of the mentioned

79)What will be the output of the following Java code?

class A

{

public int i;

private int j;

}

class B extends A

{

void display()

{

super.j = super.i + 1;

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

}

}

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) Runtime Error

**d) Compilation Error**

80) An abstract class with 100% abstract methods is equivalent to \_\_\_\_\_

a) Concrete class

b) Virtual Class

**c) Interface**

d) All the above

81) Which of these can be used to fully abstract a class from its implementation?

a) Objects

b) Packages

**c) Interfaces**

d) None of the Mentioned

82) Which of these access specifiers can be used for an interface?

**a) Public**

b) Protected

c) private

d) All of the mentioned

83) Which of these keywords is used by a class to use an interface defined previously?

a) import

b) Import

**c) implements**

d) Implements

84) Which of the following is the correct way of implementing an interface salary by class manager?

a) class manager extends salary {}

**b) class manager implements salary {}**

c) class manager imports salary {}

d) none of the mentioned

85) A java interface can contain ————

a) public static Final Variables only

b) public Abstract methods

c) Abstract methods(unimplemented) and implemented methods both

**d) public static Final Variables and abstract methods both**

86) which of the following is true about methods in an interface in java?

**a) An interface can contain only abstract method.**

b) We can define a method in an interface

c) Private and protected access modifiers can also be used to declare methods in interface

d) None

87) Which one is correct declaration for implementing two interfaces?

Consider, Interface A and B. class C wants to implements both interfaces.

**a) class C implements A, B**

b) class C implements A, implements B

c) class C implements A extends B

88) What happens when we access the same variable defined in two interfaces implemented by the same class?

a) Compilation failure

b) Runtime Exception

c) The JVM is not able to identify the correct variable

**d) The interfaceName.variableName needs to be defined**

89) What happens when a constructor is defined for an interface?

**a) Compilation failure**

b) Runtime Exception

c) The interface compiles successfully

d) The implementing class will throw exception

90) What will happen if we provide concrete implementation of method in interface?

a) The concrete class implementing that method need not provide implementation of that method

b) Runtime exception is thrown

**c) Compilation failure**

d) Method not found exception is thrown

91) What does an interface contain?

a) Method definition

**b) Method declaration**

c) Method declaration and definition

d) Method name

92) An interface in Java is like a 100% \_\_\_\_.

**a) abstract class**

b) public class

c) inner class

d) anonymous class

93) What is the output of the below Java program with an Interface?

interface Car

{

int basePrice=1000;

}

public class InterfaceTest2 implements Car

{

void changePrice()

{

basePrice = 2000;

System.out.print(basePrice);

}

public static void main(String[] args)

{

new InterfaceTest2().changePrice();

}

}

a) 1000

b) 2000

**c) Compiler error**

d) None of the above

94) Which of the following OOP concept binds the code and data together and keeps them secure from the outside world?

a) Polymorphism

b) Inheritance

c) Abstraction

**d) Encapsulation**

95)Which of the following variable violates the definition of encapsulation?

a) Array variables

b) Local variables

**c) Global variables**

d) Public variables

96)How can the concept of encapsulation be achieved in the program?

**a) By using the Access specifiers**

b) By using the concept of Abstraction

c) By using only private members

d) By using the concept of Inheritance

97)The concept of encapsulation helps in writing which type of classes in the Java programming language?

a) Abstract classes

b) Wrapper classes

c) Mutable classes

**d) Immutable classes**

98)Encapsulation is\_\_\_\_\_?

a) technique of combining more than one member functions into a single unit.

b) mechanism of combining more than one data member into a single unit.

**c) mechanism of combining more than one data members and member functions that implement on those data members into a single unit**

d) technique of combining more than one data members and member functions into a single unit, which can manipulate any data.

99) Using the concept of encapsulation security of the data is \_\_\_\_\_\_\_\_\_\_\_

**a) Ensured to some extent**

b) Purely ensured

c) Not ensured

d) Very low

100) Consider the following Java program and select the right option from the given options.

class marksofstudent

{

int subjectmarks;

public : int\* fun()

{

return &subjectmarks;

}

};

main()

{

marksofstudent s;

int \*ptr = c.fun() ;

return 0;

}

**a) The above program violates the feature of encapsulation**

b) The above program may result in undesirable conditions

c) The above program will generate an error

d) The above program is good to go

101) Which polymorphism behavior do you see in below class?

class Paint {

// all methods have same name

public void Color(int x) {

}

public void Color(int x, int y) {

}

public void Color(int x, int y, int z) {

}

}

**a) Method overloading**

b) Constructor overloading

c) Method overriding

d) Run time polymorphism

102) Which polymorphism concept is applied to inheritance relationship in java programming?

a) Method overloading

b) Constructor overloading

**c) Method overriding**

d) None

103) Which feature comes under compile time polymorphism?

a) Method overloading

b) Constructor overloading

c) Method overriding

**d) Both A and B**

104) Which statements are true

a) Overriding can be achieved with static methods

b) Final marked methods can be overridden

c) Overloading is runtime.

**d) Return type can change in case of Overloading.**

105)Which statements are true?

a) In case of overloading, binding of objects with methods happens at runtime.

**b) In the case of Dynamic polymorphism, method behavior is decided at runtime.**

c) In the case of Overriding, the creation of objects happen at compile time and these objects are used for calling objects at runtime.

d) In the case of Overriding, it is the responsibility of the compiler to bind the method calls with the method body.