OOP 2025 Quiz Questions

1. Which of the following is NOT an object-oriented programming language?

a) Java

b) C

c) C++

d) Python

2. Which of the following is a correct identifier in Java?

a) 4321

b) HelloWorld

c) ###

d) 1K

3. Which of the following is a keyword in Java?

a) UIC

b) if

c) FST

d) Love

4. Which of the following is an integer in Java?

a) “CST”

b) “DS”

c) 123

d) “AI”

5. Which of the following is NOT used for flow control in Java?

a) if-else

b) do-while

c) print

d) for

6. In the context of object-oriented programming (OOP), what is a class?

a) An instance of an object

b) A blueprint or template for creating objects

c) A specific method within an object

d) A data type representing a single value

7. What is the purpose of the `new` operator in Java?

a) To declare a variable

b) To create an object from a class

c) To define a method

d) To initialize an instance variable

8. A constructor in Java is:

a) A method that returns a value

b) A special method used to create and initialize an object

c) A type of instance variable

d) Used to define relationships between classes

9. What is the purpose of a method in the context of object-oriented programming?

a) To store data

b) To define the class structure

c) To define the behaviour or actions of an object

d) To connect different classes

10. In the example `Person` class, what would `person1.getName()` do?

a) Set the name of `person1`

b) Create a new `Person` object named `getName`

c) Return the name of the `person1` object

d) Print the age of `person1`

11. Method overloading in Java allows:

a) Multiple methods with the same name and parameters.

b) A single method to handle different numbers of arguments.

c) Multiple methods with the same name and parameter lists.

d) Methods to be called only from within the same class.

12. Which access modifier provides the *least* restrictive access to a class member?

a) `private`

b) `protected`

c) `default`

d) `public`

13. A `static` method in Java:

a) Requires an object instance to be called.

b) Cannot access instance variables.

c) Is always private.

d) Is defined outside of any class.

14. The `protected` access modifier allows access to a class member by:

a) Only the class itself.

b) The class itself, subclasses, and classes in the same package.

c) Only subclasses of the class.

d) Any class, regardless of package.

15. The `import` statement in Java is used to:

a) Create a new class.

b) Access classes from other packages.

c) Define a method.

d) Declare a variable.

16. Which key word is used to describe an inheritance from a child class to a parent class

(a) Child
(b) Extends
(c) Inherit
(d) protected

17. Which of the following sentences is correct?

(a) A child class can inherit all the instance variables from its parent class.
(b) A child class cannot inherit private instance variables from its parent class.
(c) Only protected instance variables of the parent class can be inherited by its child classes.
(d) A child class cannot declare its own instance variables.

18. How can the constructor of the Person class be called in the constructor of the Student class?

```
public class Student extends Person {
    private String school;
    public Student(String name, int age, String school) {
        _____ //call the constructor of the parent class
        this.school = school;
    }
```

A. Person(name, age)
B. super(name, age);
C. this.super(name, age);
D. none of above

19. A child class

A. can have one parent class only
B. cannot have its own constructors
C. cannot have its own private methods
D. can have two parent classes

20. A protected method in a child class

A. can override a method in parent class which is public
B. can call a private method in parent class
C. can override a method in parent class which is protected
D. None of the above

21. What does the final modifier mean?
    A. If the modifier final is placed before the definition of a *variable*, the value of this variable cannot be changed.
    B. If the modifier final is placed before the definition of a *method*, then that method cannot be redefined (overridden) in a subclass.
    C. If the modifier final is placed before the definition of a *class*, then that class cannot have subclasses.
    D. All of above

22. What is correct for the class Object?
    A. Every class is a descendent of the class Object
    B. If a class is defined without explicitly deriving from another class, it is still automatically a derived class of the class Object
    C. toString and equals are two methods in class Object
    D. All of above

23. What is the correct statement about the toString method?
    A. toString is a method that converts an integer to a string.
    B. toString is a method defined in the class Object.
    C. toString cannot be overridden.
    D. none of above.

24. What is correct about the method equals?
    A. The equals method does not have to have a parameter.
    B. This method cannot be overridden.
    C. This method is used to compare two objects.
    D. The equals method cannot be inherited.

25. When we use type casting to an object,
    A. the object itself will be changed.
    B. the type of the object cannot be changed to the type of its super class.
    C. the type of the object cannot be changed to the type of its sub class.
    D. the type of the object can be changed to the type of either its super class or its sub class.

26. Which key word is used to describe an abstract class?
    A. final
    B. abstract
    C. private
    D. Static

27. Which of the following correctly declares an abstract method?

A. public abstract int sum()
B. public abstract int sum();
C. public abstract int sum(){};
D. private abstract int sum();

28. Given the abstract class below, which methods must a non-abstract subclass override?

```
public abstract class Car {
    protected String brand;
    public int getBrand() {return brand;}
    public abstract double getSpeed();
}
```

A. getBrand()
B. getSpeed()
C. getBrand() and getSpeed()
D. The subclass does not have to override any methods.

29. Which of the following can an interface have?

A. An instance variable
B. A constructor
C. A private method
D. None of the above

30. If a class C inherits an abstract class B and implements an interface D without implementing all their abstract methods, how should class C be declared?

A. class C extends B implements D
B. class C implements D extends B
C. abstract class C extends B implements D
D. abstract class C implements D extends B

31. Which of the following is true about the try block in a try-catch-finally construct?

A. The try block is mandatary.
B. Multiple try blocks may co-exist within a single construct.
C. The try block always throws an exception.
D. None of the above

32. Which of the following is true about the catch block in a try-catch-finally construct?

A. Multiple catch blocks may co-exist within a single construct.
B. More than one catch blocks may be executed.
C. The order of the catch blocks does not matter.
D. None of the above

33. Which of the following is true about the finally block in a try-catch-finally construct?

A. The finally block is mandatary.
B. Multiple finally blocks may co-exist within a single construct.
C. The finally block is always executed.
D. None of the above

34. What is the console output for the following code?

```
try {
      System.out.print(1/0);
      System.out.print("A");
} catch(ArithmeticException e){
      System.out.print("B");
} catch(Exception e){
      System.out.print("C");
} finally {
      System.out.print("D");
}
System.out.print("E");
```

A. “ABCDE”
B. “BDE”
C. “CDE”
D. "BD”

35. Which of the following correctly declares a method capable of throwing both IOException and SQLException?

A. public void sum() throws IOException SQLException;
B. public void sum() throw IOException, SQLException;
C. public void sum() throw new IOException, SQLException;
D. public void sum() throws IOException, SQLException;

36. What is the index of the last element in an arraylist called *student*?
    a) student.length
    b) student.length-1
    c) student.size()
    d) student.size()-1

37. Which of these is an incorrect array declaration?
    a) int arr[] = new int[5]
    b) int [] arr = new int[5]
    c) int arr[] = new int[n]
    d) int arr[] = int [5] new

38. What will be the output of the following Java code?

```
1. class multidimention_array
2.     {
3.         public static void main(String args[])
4.         {
5.             int arr[][] = new int[3][];
6.             arr[0] = new int[1];
7.             arr[1] = new int[2];
8.             arr[2] = new int[3];
9.     int sum = 0;
10.            for (int i = 0; i < 3; ++i)
11.                for (int j = 0; j < i + 1; ++j)
12.                    arr[i][j] = j + 1;
13.            for (int i = 0; i < 3; ++i)
14.                for (int j = 0; j < i + 1; ++j)
15.                    sum + = arr[i][j];
16.            System.out.print(sum);
17.        }
18.    }
```

    a) 8
    b) 9
    c) 10
    d) 11

39. Which of the following statements is true about Generics in Java?
    a) Generics can only be used with arrays.
    b) Generics allow parametric polymorphism.
    c) Generics can be used only with classes and cannot be applied to methods.
    d) Generics are not type-safe at compile time.

40. What will be the output of the following Java code?

```
1. import java.util.*;
2.       public class genericstack <E>
3.       {
4.             Stack <E> stk = new Stack <E>();
5.  public void push(E obj)
6.             {
7.                   stk.push(obj);
8.  }
9.  public E pop()
10.            {
11.                  E obj = stk.pop();
12.                 return obj;
13.           }
14.      }
15.      class Output
16.      {
17.            public static void main(String args[])
18.            {
19.                  genericstack <String> gs1 = new
    genericstack<String>();
20.                  gs1.push("Hello");
21.                  System.out.print(gs1.pop() + " ");
22.                  genericstack <Integer> gs2 = new
    genericstack<Integer>();
23.                  gs2.push(2025);
24.                  System.out.println(gs2.pop());
25.           }
26.      }
```

a) Hello
b) 2025
c) Hello 2025
d) 2025 Hello