1) What is the output of the following code?

```
public class Test1 {
  public static void main(String[] args) {
    ChildClass c = new ChildClass();
    c.print();
  }
}

class ParentClass {
  int id = 1;
  void print() {
    System.out.println(id);
  }
}

class ChildClass extends ParentClass {
  int id = 2;
}
```

A. 0
B. 1
C. 2
D. Nothing

2) Suppose you create a class Square to be a subclass of GeometricObject. Analyze the following code:

```
class Square extends GeometricObject {
  double length;

  Square(double length) {
    GeometricObject(length);
  }
}
```

A. The program compiles fine, but you cannot create an instance of Square because the constructor does not specify the length of the Square.
B. The program has a compile error because you attempted to invoke the GeometricObject class's constructor illegally.
C. The program compiles fine, but it has a runtime error because of invoking the Square class's constructor illegally.

3) Analyze the following code: (**Multiple Choice)**

```
public class A extends B {
}

class B {
  public B(String s) {
  }
}
```

A. The program has a compilation error because A does not have a default constructor.
B. The program has a compilation error because the default constructor of A invokes the default constructor of B, but B does not have a default constructor.
C. The program would compile fine if you add the following constructor into A: A(String s) { }
D. The program would compile fine if you add the following constructor into A: A(String s) { super(s); }

4) Analyze the following code: (**Multiple Choice**)

```
public class Test extends A {
  public static void main(String[] args) {
    Test t = new Test();
    t.print();
  }
}

class A {
  String s;

  A(String s) {
    this.s = s;
  }

  public void print() {
    System.out.println(s);
  }
}
```

A. The program does not compile because Test does not have a default constructor Test().
B. The program has an implicit default constructor Test(), but it cannot be compiled, because its super class does not have a default constructor. The program would compile if the constructor in the class A were removed.
C. The program would compile if a default constructor A(){ } is added to class A explicitly.
D. The program compiles, but it has a runtime error due to the conflict on the method name print.

5) What is the output of running class C?

```
class A {
  public A() {
    System.out.println(
      "The default constructor of A is invoked");
  }
}

class B extends A {
  public B() {
    System.out.println(
      "The default constructor of B is invoked");
  }
}

public class C  {
  public static void main(String[] args) {
    B b = new B();
  }
}
```

A. Nothing displayed
B. "The default constructor of B is invoked"
C. "The default constructor of A is invoked""The default constructor of B is invoked"
D. "The default constructor of B is invoked""The default constructor of A is invoked"
E. "The default constructor of A is invoked"

6) What is the output of the following code?

```
public class Test {
  public static void main(String[] args) {
    B b = new B();
    b.m(5);
    System.out.println("i is " + b.i);
  }
}

class A {
  int i;

  public void m(int i) {
    this.i = i;
  }
}

class B extends A {
  public void m(String s) {
  }
}
```

A. The program has a compilation error, because m is overridden with a different signature in B.
B. The program has a compilation error, because b.m(5) cannot be invoked since the method m(int) is hidden in B.
C. The program has a runtime error on b.i, because i is not accessible from b.
D. The method m is not overridden in B. B inherits the method m from A and defines an overloaded method m in B.

7) The getValue() method is overridden in two ways. Which one is correct?

I:

```
public class Test {
  public static void main(String[] args)
    A a = new A();
    System.out.println(a.getValue());
  }
}

class B {
  public String getValue() {
    return "Any object";
  }
}

class A extends B {
  public Object getValue() {
    return "A string";
  }
}
```

II:

```
public class Test {
  public static void main(String[] args) {
    A a = new A();
    System.out.println(a.getValue());
  }
}

class B {
  public Object getValue() {
    return "Any object";
  }
}

class A extends B {
  public String getValue() {
    return "A string";
  }
}
```

A. I
B. II
C. Both I and II
D. Neither

8) Which of the following statements are true? (**Multiple Choice**)

A. To override a method, the method must be defined in the subclass using the same signature and compatible return type as in its superclass.
B. Overloading a method is to provide more than one method with the same name but with different signatures to distinguish them.
C. It is a compilation error if two methods differ only in return type in the same class.
D. A private method cannot be overridden. If a method defined in a subclass is private in its superclass, the two methods are completely unrelated.
E. A static method cannot be overridden. If a static method defined in the superclass is redefined in a subclass, the method defined in the superclass is hidden.

9) Which of the following statements are true? (**Multiple Choice**)

A. A method can be overloaded in the same class.
B. A method can be overridden in the same class.
C. If a method overloads another method, these two methods must have the same signature.
D. If a method overrides another method, these two methods must have the same signature.
E. A method in a subclass can overload a method in the superclass.

10) Given the following classes and their objects:

```
class C1 {};
class C2 extends C1 {};
class C3 extends C1 {};

C2 c2 = new C2();
C3 c3 = new C3();
```

Analyze the following statement:

```
c2 = (C2)((C1)c3);
```

A. c3 is cast into c2 successfully.
B. You will get a runtime error because you cannot cast objects from sibling classes.
C. You will get a runtime error because the Java runtime system cannot perform multiple casting in nested form.
D. The statement is correct.

11) Given the following code:

A. c1 instanceof C1
B. c2 instanceof C1
C. c3 instanceof C1
D. c4 instanceof C2

```
class C1 {}
class C2 extends C1 { }
class C3 extends C2 { }
class C4 extends C1 {}

C1 c1 = new C1();
C2 c2 = new C2();
C3 c3 = new C3();
C4 c4 = new C4();
```

12) Polymorphism means _____.

A. that data fields should be declared private.
B. that a class can extend another class.
C. that a variable of supertype can refer to a subtype object.
D. that a class can contain another class.

13) Encapsulation means _____.

A. that data fields should be declared private.
B. that a class can extend another class.
C. that a variable of supertype can refer to a subtype object.
D. that a class can contain another class.

14) Inheritance means _____.

A. that data fields should be declared private.
B. that a class can extend another class.
C. that a variable of supertype can refer to a subtype object.
D. that a class can contain another class.

15) Composition means _____.

A. that data fields should be declared private.
B. that a class extends another class.
C. that a variable of supertype refers to a subtype object.
D. that a class contains a data field that references another object.

16) Which statements are most accurate regarding the following classes?

```
class A {
  private int i;
  protected int j;
}

class B extends A {
  private int k;
  protected int m;

  // some methods omitted
}
```

A. In the class B, an instance method can only access i, j, k, m.
B. In the class B, an instance method can only access j, k, m.
C. In the class B, an instance method can only access j, m.
D. In the class B, an instance method can only access k, m.

17) What modifier should you use on the members of a class so that they are not accessible to another class in a different package, but are accessible to any subclasses in any package?

A. public
B. private
C. protected
D. Use the default modifier.

18) The visibility of these modifiers increases in this order:

A. private, protected, none (if no modifier is used), and public.
B. private, none (if no modifier is used), protected, and public.
C. none (if no modifier is used), private, protected, and public.
D. none (if no modifier is used), protected, private, and public.

19) What modifier should you use on a class so that a class in the same package can access it but a class in a different package cannot access it?

A. public
B. private
C. protected
D. Use the default modifier.

20) A class design requires that a particular member variable must be accessible by any subclasses of this class, but otherwise not by classes which are not members of the same package. What should be done to achieve this?

A. The variable should be marked public.
B. The variable should be marked private.
C. The variable should be marked protected.
D. The variable should have no special access modifier.
E. The variable should be marked private and an accessor method provided.

21) Given two reference variables t1 and t2, if t1.equals(t2) is true, t1 == t2 _____.

A. is always false
B. is always true
C. may be true or false

22) Given two reference variables t1 and t2, if t1 == t2 is true, t1.equals(t2) must be _____.

A. false
B. true

23) The equals method is defined in the Object class. Which of the following is correct to override it in the String class?

A. public boolean equals(String other)
B. public boolean equals(Object other)
C. public static boolean equals(String other)
D. public static boolean equals(Object other)

24) Which of the following statements are true?

A. Override the toString method defined in the Object class whenever possible.
B. Override the equals method defined in the Object class whenever possible.
C. A public default no-arg constructor is assumed if no constructors are defined explicitly.
D. You should follow standard Java programming style and naming conventions. Choose informative names for classes, data fields, and methods.

25) برای کلاس User یک سازنده با دو آرگمان بنویسید و متد های toString و equals را override کنید.

```
public class User {
	private String username;
	private String Password;

	@Override
	public String toString() {



	}

	@Override
	public boolean equals(                ) {

	}

}
```

26) با توجه به تعریف سه کلاس Person, Student, UndergraduateStudent تعیین کنید که کد زیر چه مقادیری را چاپ می کند؟

```
public class Main {
	public static void main(String[] args) {
		Person underGStudent = new UndergraduateStudent();
		underGStudent.method1();
		System.out.println("HEHEH It's Hard I Know!");
		underGStudent.method2();
		System.out.println("Hahaha!");
		underGStudent.method3();

	}
}
```

```java
public class Person {
	public void method1(){
		System.out.println("Person 1");
	}


	public void method2(){
		System.out.println(this.getClass().getName());
		this.method1();
		System.out.println(this instanceof Person);
		method1();
		System.out.println(getClass().getName());
		System.out.println("Person 2");
	}

	public void method3(){
		method1();
		System.out.println("Person 3");
	}
}
```

خروجی:

```java
public class Student extends Person {
	@Override
	public void method1(){
		super.method1();
		System.out.println("Student 1");
	}

	@Override
	public void method2(){
		super.method2();
		method1();
		this.method1();
		this.method3();
		System.out.println("Student 2");
	}

	@Override
	public void method3(){
		super.method3();
		System.out.println("Student 3");
	}
}
```

```java
public class UndergraduateStudent extends Student{
	@Override
	public void method1(){
		super.method1();
		System.out.println("Undergraduate Student 1");
	}

	@Override
	public void method2(){
		super.method2();
		System.out.println("Undergraduate Student 2");
	}
}
```