سوال اول

الف) وراثت یکی از مفاهیم مهم و کاربردی در برنامه نویسی شیگرا است.در زیر بعضی از مزایای وراثت بیان شده است:

- وراثت به استفاده مجدد از کد کمک میکند. کلاسهای فرزند میتوانند از ویژگیها و متدهای کلاس والد استفاده کنند بدون آنکه نیاز به نوشتن دوباره کدهای آن داشته باشند.
- وراثت میتواند باعث صرفهجویی در زمان شود زیرا نیاز به نوشتن مجدد کد نیست.
- وراثت یک ساختار واضح را ارائه میدهد که به راحتی قابل فهم و درک است.
- در وراثت کلاس والد میتواند بعضی از ویژگیها و متدهای خود را از دید کلاسهای فرزند پنهان کند.
- در وراثت کلاسهای پایه میتوانند متدهای کلاس والد را با توجه به ویژگیهای خود override کنند.

ب) در زبان Java یک class نمیتواند بهصورت همزمان از چند کلاس ارث بری کند زیرا ممکن است diamond problem به وجود بیاید.

مثال زیر را در نظر بگیرید که در آن Class1 و Class2 متد یکسانی به نام ()display دارند.اگر یک کلاس می توانست بهطور همزمان از دو کلاس ارث بری کند پس کلاس Test میتوانست ویژگیها و متدهای هر دو کلاس Class1 و Class2 را دارا باشد. پس در این صورت کلاس Test دو متد ()display دارد و هنگامی که یک شی از این کلاس این متد را فراخوانی کند برنامه نمیتواند تشخیص دهد کدام یک از آنها را باید فراخوانی کند.

```java
class Class1{
	public void display(){
		System.out.println("Display method inside Class1.");
	}
}

class Class2{
	public void display(){
		System.out.println("Display method inside Class2.");
	}
}

//let multiple inheritance is possible.
public class Test extends Class1, Class2{
	public static void main(String args[]){
		Test obj = new Test();
		//Ambiguity problem in method call which class display() method will be called.
		obj.display();
	}
}
```

ج) عبارت کلیدی Super در زبان Java یک متغیر مرجع است که به شی کلاس والد اشاره می‌کند.هر زمان که یک شی از کلاس فرزند ایجاد می‌شود یک شی از کلاس والد به‌طور ضمنی ایجاد می‌شود که توسط متغیر مرجع (Super) ارجاع داده می‌شود.

د) تفاوت اصلی بین abstraction و Inheritance در این است که abstraction جزییات داخلی را پنهان می‌کند و تنها عملکرد را به کاربر نشان می‌دهد در حالی که Inheritance اجازه استفاده از جزییات داخلی مانند ویژگی‌ها و متدها را می‌دهد.

ه) در جدول زیر برخی تفاوت‌های بین method overloading و method overriding ذکر شده است:

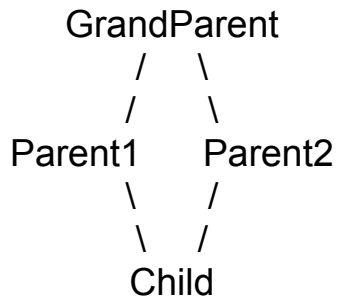
Method overriding	Method overloading
برای پیاده‌سازی دوباره متدی که قبلاً توسط کلاس والد آن ارائه شده است، استفاده می‌شود.	به خوانایی بیشتر برنامه کمک می‌کند.
همیشه به وراثت نیاز دارد.	ممکن است به وراثت نیاز داشته باشد یا نداشته باشد.
متدها باید نام و signature یکسان داشته باشند.	متدها باید نام یکسان و signature متفاوت داشته باشند.
در اینجا باید return type مشابه باشد.	در اینجا return type می تواند مشابه باشد یا نباشد اما باید پارامترها را عوض کنیم.
یک روش چند ریختی run-time است.	یک روش چندریختی compile-time است.

و) منجر به کامپایل ارور خواهد شد زیرا برنامه نمی‌تواند تشخیص دهد که کدام متد را باید فراخوانی کند.

سوال دوم

بخش اول

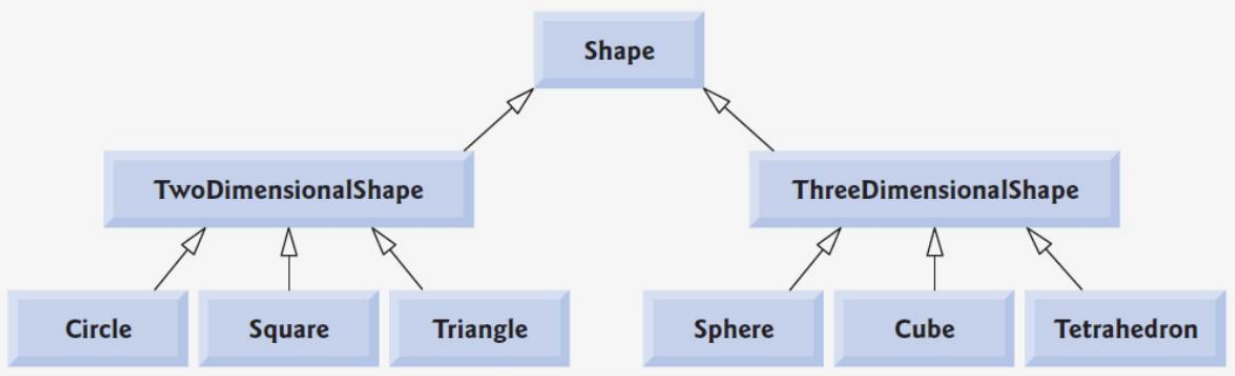
در جاوا، اگر کلاس فرزند بیشتر از یک کلاس والد داشته باشد، multiple inheritance خواهیم داشت. حال، فرض کنید که یک متد با signature یکسان در کلاس‌های والد داشته باشیم. اگر در کلاس فرزند، این متد را صدا بزنیم کامپایلر نمی‌تواند تشخیص دهد که متد کدام والد را اجرا کند. به این مشکل در جاوا multiple inheritance می‌گوییم. به مثال زیر توجه کنید:




```java
class GrandParent {
    void test() {
        System.out.println("Grandparent");
    }
}

class Parent1 extends GrandParent {
    void test() {
        System.out.println("Parent1");
    }
}

class Parent2 extends GrandParent {
    void test() {
        System.out.println("Parent2");
    }
}

class Child extends Parent1, Parent2 {
    public static void main(String[] args) {
        Child child = new Child();
        // Calling test() method from its parent classes
        // which will throw compilation error
        child.test();
```

```
    }
}
```

بخش دوم



الف) درستی یا نادرستی دستورات زیر را با ذکر دلیل مشخص کنید.

- Shape shape = new sphere(); T
- Square square = new TwoDimensionalShape(); T
- ThreeDimensionalShape tds = new Triangle(); F
- ThreeDimensionalShape tds = new Cube(); T
- Tetrahedron tetrahedron = new Shape(); F

ب) ابتدا دستورات زیر را در نظر بگیرید :

- Circle c1 = new Circle();
- TwoDimensionalShape t1 = new Triangle();
- Shape s1 = new Shape();
- ThreeDimensionalShape t2 = new ThreeDimensionalShape();
- Shape s2 = new TwoDimensionalShape();
- Shape s3 = new Cube();
- Sphere s4 = new Sphere();
- Tetrahedron t3 = new Tetrahedron();

حال درستی یا نادرستی دستورات زیر را با ذکر دلیل مشخص کنید.

- c1 = t1; F
- s1 = t2; T
- s2 = t3; T
- t2 = s3; T
- s4 = t3; F
- t2 = t1; F

سوال سوم

الف) جای خالی را با عبارت مناسب پر کنید.

a) PROTECTED
b) CONSTRUCTOR
c) SUPER
d) PRIVATE
e) POLYMORPHISM
f) ABSTRACT
g) ABSTRACT
h) INTERFACE
i) PRIVATE
j) CONCRETE

ب) صحیح یا غلط بودن عبارات زیر را مشخص کنید.

1) F
2) T
3) T
4) T
5) F
6) F
7) F