«باسمه تعالی»

سوال 1:

قسمت الف) درستی یا نادرستی عبارات زیر را با ذکر دلیل برای عبارات نادرست ، تعیین کنید.

1) **در جاوا می توانیم آدرس ذخیره شدن یک شیء را مشخص کنیم.**

نادرست- برخلاف زبان C ما رد جاوا به آدرس ها دسترسی مستقیم نداریم و وقتی به کمک جاوا چیزی را در حافظه ذخیره میکنیم نمی دانیم در چه آدرسی ذخیره میشود و با هربار اجرای برنامه آدرس دهی ها تغییر میکند.

2) **اگر access modifier کلاسی public نباشد ، نمی توان از آن کلاس ارث بری کرد.**

نادرست- اگر accesss modifier کلاسی به صورت public یا protected باشد میتوان از آن ارث بری کرد. اما اگر access modifier آن به صورت private تعریف شود نمیتوان دیگر از آن کلاس ارث بری کرد.

3) **در یک کلاسfield هایی که protected تعریف شدهاند ، با استفاده از ارث بری، در خارج packageشان قابل دسترسی اند.**

درست

4) **اگر تابع override شده در subclass ، publicباشد، میتوان نتیجه گرفت در superclassهم publicاست**

نادرست- اگر یک تابع در superclass داریم و در subclass میخواهیم override کنیم باید access modifier آن در subclass بیشتر یا مساوی با access modifier تعریف شده در superclass باشد. بنابراین اگر تابع override شده در subclass ، publicباشد، در superclass میتواند protected هم باشد. (حواسمان هست که اگر تابع موجود در والد private باشد در subclass دیگر override نمیشود.)

5) **متدهای private را می توانیم override کنیم.**

نادرست- متدهای private را override نمی توان کرد زیرا اصلا برای کلاس های دیگر قابل دسترس و مشاهده نیست. درواقع اگر این کار ار بکنیم انگار یک متد جدید داریم برای subclass تعریف میکنیم که ربطی به متد موجود در superclass ندارد.

قسمت ب)

1) همه ی کلاس ها در جاوا از کلاس **Object** ارث بری می کنند.
2) اگرfield های یک کلاس را **private**تعریف کنیم، subclassنمی تواند از آن ها ارث بری کند.
3) وقتی دو تابع در یک کلاس اسم یکسان اما پارامترهای ورودی مختلف داشته باشند ، **overloading** رخ داده است.
4) برای override کردن ی تابع، تابع نوشته شده در subclass باید **نام** و **پارامترهای** یکسان با آن تابع داشته باشد.

سوال2:

قمت اول:

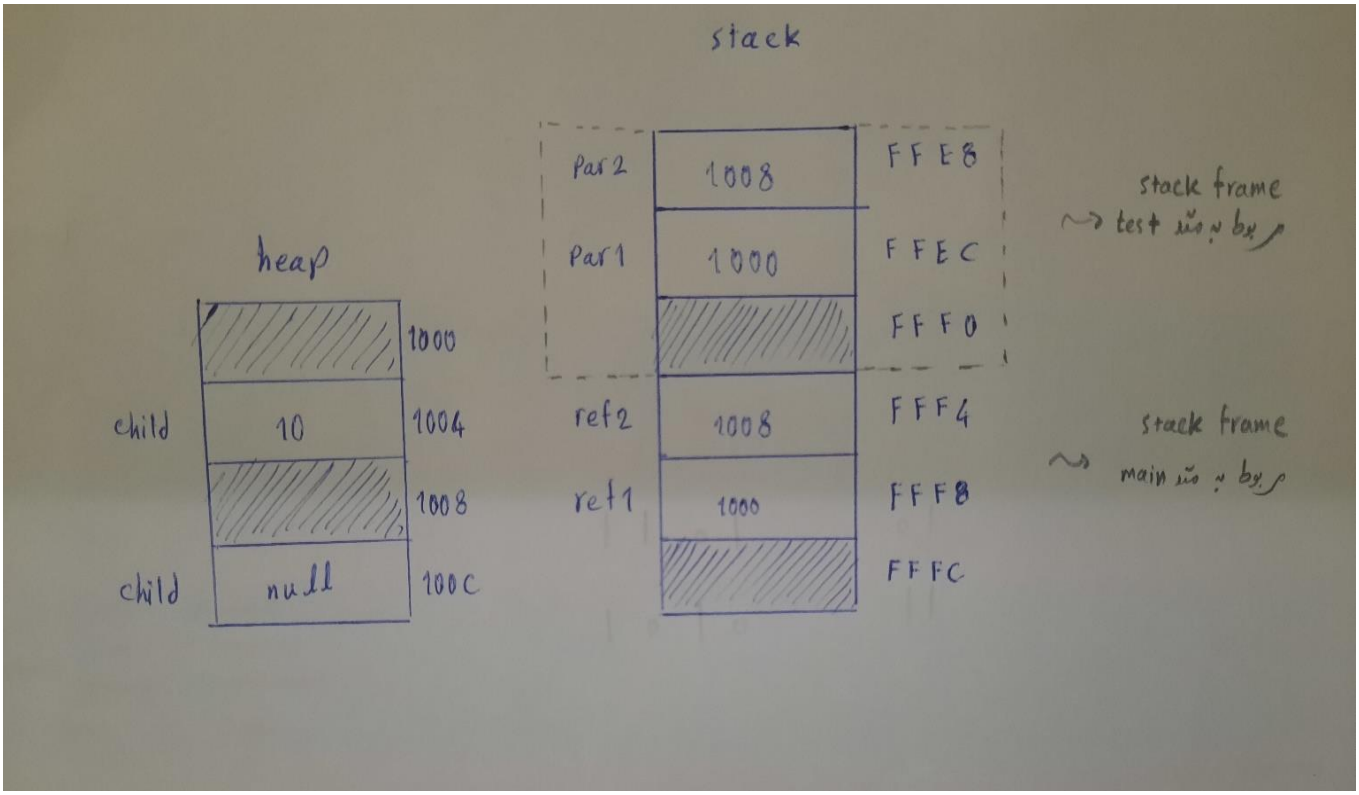
در این قطعه کد به ارورClassCastExeption برمیخوریم که یک compile_time error می باشد .

تا خط دوم صحیح می باشدو ما میتوانیم یک مقدار از جنس subclass رو داخل متغیر از جنس superclass بریزیم اما برعکس آن امکان ندارد.درسته که اکنون داخل a1 که یک مجموعه بزرگ تر می باشد b1 را قرار دادیم اما همانطور که گفتیم داخل a1 میتواند چیزهای دیگری هم باشد و اگر مقدار داخل a1 واقعا فقط b1 نباشد پس طبیعی هست که به ارور بخوریم.

یک راه که برای رفع آن می باشد این است که cast انجام دهیم و درواقع کد به این شکل شود:

```
B b1 = new B();
A a1 = b1;
b1 = (B) a1;
```

قسمت دوم:



Stack frame مربوط به متد تست به صورت موقت ایجاد میشود و پس از انجام عملیات برنامه و استفاده از آن متد نابود میشود.