

ب:

ابتدا به بررسی خط به خط کد می پردازیم

در خط اول یک Object با نوع `str1` درست می شود که در آن `CEIT` نوشته شده است و بعد با اسم `str1` اسم گذاری میشود.

در خط بعد ابتدا سمت راست مساوی باید محاسبه شود و به این دلیل که تمام `summand` ها از نوع `string` هستند عملیات `concatenation` انجام میشود.

حال باید در دو خط بعدی نمایش دهیم که `str1==str2` و `(str1.equals(str2))` چه نتیجه ای میدهد. با توجه به اینکه در زبان `java` رشته های `immutable` هستند و اگر بخواهیم یک رشته را تغییر دهیم باید از آن رشته مورد جدیدی درست کنیم؛ پس در واقع این استدلال مطرح میشود که اگر دو رشته داشته باشیم که مقدار های آنها برابر است زبان `java` سعی می کند آنها را تبدیل به یک `object` در `heap` کند که کمتر فضا بگیرد و از طرفی یک `overhead` اضافه میکند که نگاه که نیاز شد آن دو متغیر را از هم جدا کند و محاسبه های متفاوتی را بر روی آنها انجام دهد. پس وقتی ما `str1` و `str2` را میسازیم `jvm` متوجه می شود که یک `string` قرار است درست شود پس به هر دو متغیر `str1` و `str2` `reference` های برابر میدهد و پس در `==` مقدار آنها برابر خواهد بود.

از طرفی در قسمت بعدی وقتی ما از `Integer` میخواهیم که عدد ۱۲۳ را به `string` برای ما تبدیل کند دیگر `jvm` آن را به عنوان یک شیء در نظر نمی گیرد (چون از مقداری که به داخل تابع می رود خبری ندارد) پس در واقع دو `reference` جدا برای آنها میگیرد.