تمرین شماره ۱ :

a) public

b) new

c) default , zero

تمرین شماره ۲ :

الف) اینکه اسم constructor دقیقا شبیه اسم کلاس است و اینکه هیچ یک از انواع بازگشتی را ندارد.

ب) public class Student

پ) دو پارامتر دارد و نوع اولی String و نوع دومی double است

تمرین شماره ۳ :

الف) غلط : زیرا آغاز نام یک method با حرف کوچک است و تنها کلاس ها را در شروع نامشان از حرف بزرگ استفاده می شود

ب) درست

پ) درست

ت) غلط : زیرا این متغیرها محلی هستند و درون خود آن method قابل شناسایی هستند

ث) درست

ج) غلط : زیرا در واقع جاوا بر روی یک ماشین مجازی اجرا می شود و در سیستم عامل های مختلف قابل اجراست

تمرین شماره ۴ :

API : مخفف Aplication Programming Interface است که در واقع یک رابط کاربردی برنامه نویسی ست که اجازه ارتباط برنامه های دیگر با نرم افزار مورد نظر را کنترل می کند و باعث تعامل نرم افزار ها با هم میشود .

Constructor overloading : در واقع می توان بیش از یک constructor تعریف کرد که مثلا در تعداد پارامتر ها باهم تفاوت دارند . در واقع وظایف متفاوتی دارند ولی همه آن ها هم نام با کلاس هستند .

Abstraction : در واقع یک نوع ساده سازی مسآله پیش رو می باشد به گونه ای که بتوان به یک درک کلی از آن رسید.

Modularization : یعنی به این فکر کنیم که برای اجرای یک پروژه به چه ابزاری نیاز داریم و در فکر طراحی آن ابزار ها (ماژول) بر آییم .

تمرین شماره ۶ :

بزرگترین مقدار قابل محاسبه برای فیبوناچی : 1836311903

برای اینکه بتوان به صورت نامحدود دنباله فیبوناچی را محاسبه کرد می توان کلاس جدیدی برای اعداد صحیح تعریف کرد که ارقام این اعداد را در لیستی نگه می دارد و متدی تعریف نمود که بتوان این اعداد را به صورت رقم به رقم جمع یا ضرب نمود همچنین می توان عملگر های معمولی را نیز برای این کلاس تعریف مجدد کرد .

تمرین شماره ۷ :

a)int a,b,c,d ; d = a*b*c ;

b)Scanner scanner = new Scanner(System.in);

c)int x,y,z,result ;

d)System.out.println(“Pls enter first integer:”) ;

e) x = System.in ;

f)System.out.println(“Pls enter second integer:”) ;

g)y = System.in ;

h)System.out.println(“Pls enter third integer:”) ;

i)z = System.in ;

j)result = x*y*z ;

k)System.out.println(“Product is : “ + result) ;

تمرین شماره ۸ :

Fields :

private double Top speed

private double acceleration

private String owner

private double price

private boolean isLocked

private double xPos

private double yPos

methods :

public void turnToRight()

public void turnToLeft()

public void accelerate(boolean key)

private void continuousMoving()

متد accelerate متدی ست که به حرکت ماشین شتاب می دهد و ورودی این متد درواقع علامت شتاب است که می تواند صفر به معنای منفی یا ۱ به معنای شتاب مثبت باشد . متد continuosMoving متدی ست که حرکت ماشین را در همان وضعیتی که قرار دارد را ادامه می دهد و توسط کاربر استفاده نمی شود بلکه به طور خودکار اجرا می گردد (مثلا توسط یک کلاس مرکزی که دوست این کلاس تعریف می شود و حرکت اشیا را کنترل می کند اجرا می شود)

تمرین شماره ۹ مسآله 3.32 :

روش ذخیره کردن اعداد به صورت دوازده ساعته بهتر از تغییر نحوه نمایش است زیرا با این روش می توان درستی ورودی را نیز تعیین کرد و با همان اعداد کار کرد.

البته این مسآله تا حدی نیز می تواند بسته به نوع کاربرد و حتی سلیقه ای باشد!