

به نام خدا

تمرین اول درس برنامه‌نویسی پیشرفته

نیم‌سال دوم ۱۳۹۹-۱۴۰۰

۱. تمامی فایل‌های کد را به همراه فایل متنی که در قالب pdf است (مورد سوم را بخوانید) به صورت یک فایل آرشیو zip (zip!= rar) که به قالب زیر نام‌گذاری شده است، بارگذاری نمایید.

AP-HW1-FirstName_LastName-StudentNumber.zip

AP-HW1-Saman_Hoseini-9731079.zip

۲. در سوال‌هایی که ورودی و خروجی مطلوب آن‌ها مشخص شده است، برنامه‌ی شما به صورت ماشینی تصحیح می‌شود. بنابراین رعایت نحوه ورودی‌گرفتن و نمایش خروجی اهمیت بسیاری دارد. دقیقاً همان‌طور که از شما خواسته شده است ورودی‌ها را خوانده و خروجی‌ها را تولید کنید.

۳. پاسخ سوالات تشریحی را به صورت تایپ‌شده و در قالب یک فایل pdf (برای کل تمرین) تحویل دهید.

۴. در صورت مشاهده هرگونه تقلبی، نمره منفی ۱۰۰ برای افراد متقلب لحاظ می‌گردد.

۵. در صورت وجود هرگونه ابهام می‌توانید از طریق گروه تلگرامی با تدریس‌یاران در ارتباط باشید.

۶. سقف نمره دریافتی از تمرین با احتساب سوال امتیازی ۱۱۰ می‌باشد.

مهلت تحویل: تا جمعه ۲۲ اسفند ۱۳۹۹ ساعت ۲۳:۵۵ شب

- سوال اول..... ۳
- سوال دوم..... ۴
- سوال سوم..... ۵
- سوال چهارم..... ۶
- سوال پنجم..... ۷
- سوال ششم..... ۹
- سوال هفتم (امتیازی)..... ۱۱

سوال اول

به سوالات زیر پاسخ دهید.

الف) تفاوت بین JRE، JDK و JVM چیست؟ توضیح دهید.

ب) یک برنامه‌ی جاوا چه روندی را می‌گذراند تا اجرا (run) شود؟ دیاگرام آن را رسم کنید.

ج) نوع‌های ابتدایی (primitive data types) چه هستند؟ مختصر توضیح دهید و چند نمونه از آن‌ها را نام ببرید.

د) به انواع مختلف دیدگاه در طراحی و برنامه نویسی نرم افزارها اصطلاحاً پارادایم (Paradigm) گفته می‌شود. در درس مبانی برنامه نویسی با نحوه‌ی برنامه نویسی ساخت یافته (Structured Programming) آشنا شدید و در درس برنامه نویسی پیشرفته با دیدگاه شی گرای (Object Oriented Programming) آشنا خواهید شد. درباره‌ی پارادایم‌های Object Oriented، Functional و Structural تحقیق کنید. تفاوت‌ها و ویژگی‌های هر کدام را توضیح دهید.

ه) هر کدام از مفاهیم زیر را به صورت مختصر توضیح دهید:

Class, Object, Constructor, Method, Parameter, Instance

و) با مفهوم تابع (function) در درس مبانی برنامه نویسی آشنا شدید. تفاوت method و function در چیست؟

سوال دوم

صحیح یا غلط بودن عبارات زیر را مشخص کنید.

1. Java programming is not statically-typed, means all variables should not first be declared before they can be used.
2. A .class file contains bytecodes.
3. An array in the Java programming language has the ability to store many different types of values.
4. You are able to do arithmetic operations on a reference variable (increment it for example) in Java as well as C.
5. Java is 100% object oriented.

سوال سوم

جاهای خالی را پر کنید.

- A. Java classes contain (which implement class behaviors) and (which implement class/object data).
- B. In Java, the unit of programming is the, from which are eventually instantiated.
- C. Java programmers concentrate on creating their own user-defined types, called
- D. The command of JRE executes an application.
- E. The command of JDK compiles a Java program.
- F. A Java program file must end with the file extension.
- G. When a Java program is compiled, the file produced by the compiler ends with the file extension.
- H. The file produced by the Java compiler contains that are interpreted to execute a Java applet or application.

سوال چهارم

کد زیر یک الگوریتم مرتب سازی (Sort) است ولی مشکلاتی دارد و خروجی غلط می دهد. غلط های آن را اصلاح کنید:

```
class FindMistake {
    private static void printArray(int arr) {
        int n = arr.length;
        for (int i = 0; i <= n; ++i)
            System.out.println(arr[i] + " ");

        System.out.println();
    }

    private static void sort(int arr[]) {
        int n = arr.length;
        for (int gap = n / 2; gap >= 0; gap /= 1) {
            for (int i = gap; i < n; i += 1) {
                int temp = arr[i]

                int j;
                for (j = i; j >= gap && arr[j - gap] > temp; j -= gap)
                    arr[j] = arr[j + gap];

                arr[j] = temp;
            }
        }
    }

    public static void main(String args[]) {
        int arr[] = { 12, 34, 54, 2, 3 };

        System.out.println("Array before sorting");

        printArray(arr);
        sort(arr);

        System.out.println("Array after sorting");

        printArray(arr);
    }
}
```

سوال پنجم

برنامه‌ای بنویسید که چهار عمل اصلی (جمع، تفریق، ضرب، تقسیم) را برای اعداد مختلط انجام دهد.
اعداد مختلط دارای دو بخش حقیقی و موهومی هستند. برای مطالعه بیشتر به لینک زیر مراجعه کنید:

https://en.wikipedia.org/wiki/Complex_number

نحوه ورودی گرفتن در جاوا:

<https://www.geeksforgeeks.org/scanner-class-in-java/>

ورودی:

در خط اول ورودی دو عدد a و b داده می شود که a بخش حقیقی و b ضریب بخش موهومی عدد مختلط اول است.
در خط دوم دو عدد دیگر داده میشود که بیانگر عدد مختلط دوم است.
در خطهای بعدی تا زمانی که کاراکتر $\#$ را مشاهده نکرده اید عملگر ورودی بگیرید.

خروجی:

با توجه به عملگرها، عملیات ریاضی را بین دو عدد مختلط ورودی به ترتیب انجام دهید و خروجی را چاپ کنید.

نمونه ورودی خروجی اول:

Input1:

2 -1

-3 2

*

+

-

#

Output1:

-4+7i

-1+i

5-3i

نمونه ورودی خروجی دوم:

Input2:

-1 1

1 1

*

-

#

Output2:

-2+0i

-2+0i

سوال ششم

می‌خواهیم برنامه‌ای بنویسیم که یک رشته را با استفاده از یک کلید ۸ بیتی رمزنگاری کند. برای نوشتن انکودر اول باید یک کلید ۸ بیتی به عنوان ورودی بگیرد. سپس یک رشته ورودی می‌گیرد و آن را به صورت باینری می‌نویسد. (برای انجام این کار کد ASCII کاراکترها را به یک عدد باینری ۸ بیتی تبدیل کنید) در انتها هر کاراکتر باینری شده را با کلید داده شده باید XOR کند.

XOR یک عملیات منطقی مانند (AND, OR) است با این تفاوت که تنها زمانی درست (true) می‌شود که ورودی‌ها متفاوت باشند.

جدول ارزشی XOR:

A	B		A XOR B
0	0		0
0	1		1
1	0		1
1	1		0

ورودی:

در خط اول یک رشته ۸ کاراکتری به عنوان کلید می‌گیرد.

در خط دوم یک رشته برای رمزنگاری می‌گیرد.

خروجی:

رمزنگاری شده رشته ورودی را به صورت باینری خروجی می‌دهد. (رشته‌ای به فرم باینری)

Input:

11110011

Java

Output:

10111001 10010010 10000101 10010010

توضیح:

ابتدا ورودی Java را به صورت باینری نوشته:

```
01001010 01100001 01110110 01100001
```

حال بیت به بیت با کلید آن را XOR می‌کنیم و در آخر، حاصل همان خروجی رمزنگاری شده است.

```
01001010 01100001 01110110 01100001  
(+) 11110011 11110011 11110011 11110011  
-----  
10111001 10010010 10000101 10010010
```

سوال هفتم (امتیازی)

برنامه ای بنویسید که مشخص کند آیا یک رشته می تواند دقیقاً در یک ویرایش به رشته دیگری تبدیل شود یا خیر. دو رشته به برنامه خود بدهید و مشخص کنید آیا با فقط ۱ ویرایش می شود رشته اول را به رشته دوم تبدیل کند یا خیر، ویرایش ها می تواند حذف کردن، اضافه کردن و یا جایگزین کردن یک کاراکتر باشد.

ورودی:

دو رشته در دو خط ورودی داده می شود.

خروجی:

در صورت امکان تبدیل رشته اول به رشته دوم تنها با یک تغییر مقدار True و در غیر این صورت False خروجی داده شود.

مثال:

Input:

xyz

xz

Output:

True

توضیح: با فقط یک ویرایش می توان دومی را بدست آورد. (حذف کردن کاراکتر Y)

Input:

xyz

xyyz

Output:

True

توضیح: با فقط یک ویرایش می توان دومی را بدست آورد. (اضافه کردن کاراکتر Y)

Input:

xzyz

xxyx

Output:

True

توضیح: با فقط یک ویرایش می‌توان دومی را بدست آورد. (کاراکتر Z تماماً با کاراکتر X جایگزین شده‌است)

Input:

xyz

xxx

Output:

False

توضیح: تعداد ویرایش‌های مورد نیاز ۲ است.

Input:

xyz

xyz

Output:

False

توضیح: تعداد ویرایش‌های مورد نیاز ۰ است.