



# آزمون نرم افزار - تکلیف دوم

## دانشگاه صنعتی اصفهان

استاد درس: دکتر شیرین بقولی زاده

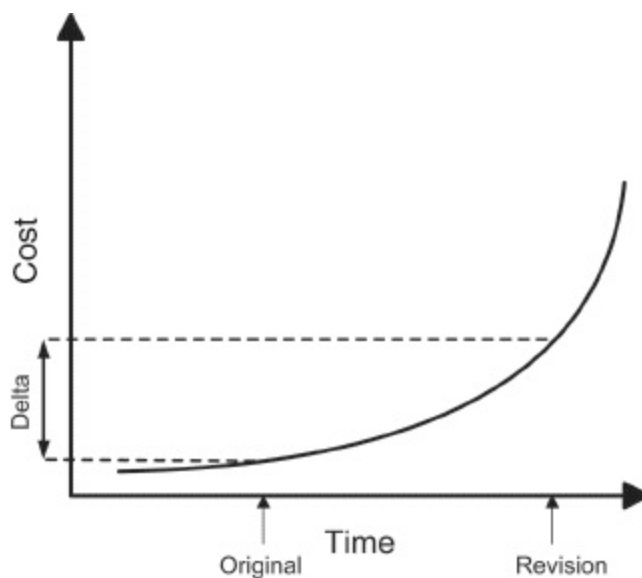
دستیاران آموزشی درس: سحر رمضانی، حدیث غفوری، رسول کامکار، حسن کبیریان، امیر ارسلان یآوری

(۲۵ نمره)

### ۱. پاسخ سوالات تشریحی زیر را بنویسید.

الف) براساس روش چابک (Agile)، چگونه می توانیم تست ها را به یک سیستم قدیمی (legacy system) موجود که هیچ تستی برای آن وجود ندارد و با محدودیت اسناد مواجه است، اضافه کنیم؟  
خطرات محتمل و روش های کاهش آنها را ذکر کنید.

ب) بر اساس نمودار زیر:



- مفهوم نمودار زیر را توضیح داده و توصیف کنید. هدف این نمودار چیست؟
- دو دلیل افزایش مداوم هزینه ها در این نمودار را نام ببرید.

---

ج) چالش‌ها یا نقاط ضعف در روش چابک که ممکن است بر کارایی تست‌ها تأثیر بگذارد را توضیح دهید. همچنین بفرمایید چگونه می‌توان نقاط ضعف روش‌های چابک در تست را مدیریت کرد؟

د) توضیح دهید چگونه داستان‌های کاربری (user stories) در روش‌های چابک در توسعه مبتنی بر تست استفاده می‌شوند.

و) مفاهیم Prefix values, Controllability, Observability, Software Testability و Postfix values را در رابطه با تست نرم‌افزار توضیح دهید.

ه) یک صفحه‌ی ورود ساده بنویسید و توضیح دهید چگونه آن را بر اساس روش data-driven تست خواهید کرد.

ی) موارد زیر را در نظر بگیرید سپس آنها بر اساس قابلیت کنترل و قابلیت مشاهده مرتب کنید و مقایسه کنید.

- نرم‌افزاری برای یک میکروکنترلر که ورودی‌ها را از یک سنسور دما گرفته و یک کولر را کنترل می‌کند.
- یک وب‌سایت با رابط کاربری تمیز و کاربر پسند.
- یک سیستم که ورودی را از صفحه‌کلید خوانده و راه‌آهن‌ها را کنترل می‌کند.

(۱۵ نمره)

۲. برای هر یک از سناریوهای زیر، unit test بنویسید.

- Test adding items to the shopping cart with different quantities.
- Test removing items from the cart with different quantities.
- Test removing items that don't exist in the cart.
- Test adding negative quantities.
- Test removing more items than is available.

۳. شما باید برای یک کلاس پایتونی به نام **LibrarySystem** که یک کتابخانه را به شکل خیلی ابتدایی مدیریت می‌کند **unit test** بنویسید. نحوه مدیریت و متدهای این کلاس و عملکرد آن‌ها در کد زیر مشخص شده است. شما باید با استفاده از کتابخانه **unittest** در پایتون یک کلاس تست ایجاد کنید و تابع‌های مختلف این کلاس را مورد تست قرار دهید.

(۲۵ نمره)

```
class LibrarySystem:
    def __init__(self):
        self.books = {}

    def add_book(self, book_id, title, author, quantity):
        if book_id in self.books:
            raise ValueError("Book with ID {} already exists".format(book_id))
        self.books[book_id] = {'title': title, 'author': author, 'quantity': quantity}

    def borrow_book(self, book_id):
        if book_id not in self.books:
            return False
        if self.books[book_id]['quantity'] == 0:
            return False
        self.books[book_id]['quantity'] -= 1
        return True

    def return_book(self, book_id):
        if book_id not in self.books:
            return False
        self.books[book_id]['quantity'] += 1
        return True

    def get_book_quantity(self, book_id):
        if book_id not in self.books:
            return None
        return self.books[book_id]['quantity']
```

۴. با توجه به کد زیر، فرض کنید خروجی قابل اجرا (executable) از آن را داریم؛ اکنون یک مورد آزمون (test case) کامل شامل پیشوند (prefix)، مقادیر آزمون (test values)، مقادیر مورد انتظار (expected values) و پسوند (postfix) برای تست تابع findPrintAVG با 10 مقدار برای scores بنویسید.

(۲۵ نمره)

```
void addScores(std::vector<float>& scores) {
    std::cout << "enter scores (enter -1 to finish):" << std::endl;
    // code for inputting scores until -1 is entered
}

void findPrintAVG(const std::vector<float>& scores) {
    // code for finding and printing the average of scores
}

int main() {
    int command = -1;
    std::vector<float> scores;
    while (command) {
        std::cin>> command;
        switch (command) {
            case 1:
                addScores(scores);
                break;
            case 2:
                findPrintAVG(scores);
                break;
            case 0:
                std::cout << "goodbye!" << std::endl; break;
                break;
            default:
                std::cout << "wrong command!" <<< std::endl;
        }
    }
}
```

---

۵. یک ابزار برای **Automated testing** (به عنوان مثال **Cypress، Appium** یا ... که در درس به آنها اشاره نشده است) را انتخاب کنید و عملکردهای آن را بررسی کنید (اجزای آن را نام برده و نحوه کار آن را توضیح دهید). (دقت فرمایید پاسخ به این سوال را باید در قالب یک صوت (voice) ارسال فرمایید؛ همچنین دقت فرمایید منابعی که برای پاسخ به این سوال استفاده کرده‌اید را نیز باید ذکر کنید).

(۱۰ نمره)

### در پاسخی به سوالات، به موارد زیر دقت فرمایید.

---

ایرادی در استفاده از دستیاران متنی هوشمند (AI chatbots) وجود ندارد اما در صورت وجود مشابهت میان پاسخ‌های هر دو فرد، نمره‌ای به هیچ‌یک اختصاص نخواهد یافت؛ بنابراین در جهت حل سوالات مجاز به استفاده خواهید بود اما در صورتی که مستقیماً جواب آنها را وارد فرمایید، نمره‌ای نخواهید گرفت.

پاسخ‌های خود را به صورت خوانا بنویسید. نهایتاً تمامی فایل‌های پاسخ را به صورت فشرده (zip یا xz) در سامانه با رعایت الگوی `HW1-std_number-first_name-last_name.file_extention` قرار دهید.

در صورت وجود هرگونه سوال و ابهام، موضوع را در گروه تلگرام  عنوان فرمایید (تعهدی در قبال پاسخ‌گویی به سوالات در قالب دیگری وجود ندارد)

---

(موفق باشید :)