

لعام الدراسي 2017 / 2018

امتحانات الفصل الثاني

جامعة دمشق

اسم المقرر : هندسة برمجيات

الاختصاص: تقنيات الحاسوب

كلية التطبيقية

العلامة: %80

السنة: الثالثة

13/6/2018

السؤال الأول: (40 علامة)

ليكن لدينا النظام S ، والذي يخدم المستخدم A1 بالخدمات u1 و u2 و u3 و يخدم المستخدم A2 بالخدمات u4 و u5. علماً أن:

- A1 يتفاعل مع النظام عن طريق الوحدات w1 و w2 ،
- بينما A2 يتفاعل مع النظام عن طريق الوحدة w3 .
- جميع الخدمات يجب أن تنتهي بزمن لا يتجاوز 30 ثانية
- يجب أن يعمل النظام على مدار 24 ساعة

المطلوب:

1. حدد المتطلبات الواردة في النص أعلاه. (10 علامات)
2. أرسم Use case Diagram . (10 علامات)
3. حدد الـ Classes . (10 علامات)
4. أرسم Sequence Diagram الذي يمثل تفاعل A1 مع النظام . (10 علامات)

السؤال الثاني: (10 علامات)

أوضح الخطوات الأساسية الواجب إتباعها في مرحلة جمع المتطلبات Requirements Gathering

السؤال الثالث: (10 علامات)

شرح بإيجاز النمذجة Modeling

السؤال الرابع: (20 علامة)

عدد الخصائص الواجب توفرها في المتطلبات مع شرح موجز لكل واحدة منها.

مع التمنيات بالنجاح

العام الدراسي 2018 / 2017  
اسم المقرر : هندسة برمجيات  
العلامة: 80%

امتحانات الفصل الثاني  
السنة الثالثة  
الاختصاص: تطبيقات الحاسوب

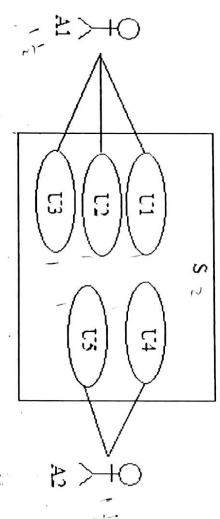
جامعة دمشق  
الكلية التطبيقية

اسم المقرر \_\_\_\_\_  
مدة دستوى \_\_\_\_\_

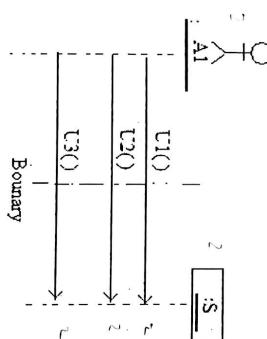
السؤال الأول: 40 علامات

1. (10 علامات)  
المطلبات الوظيفية: S, U1, U2, U3, U4, U5, U6, U7, U8, U9, U10.  
المطلبات الغير وظيفية: جميع الخدمات يجب أن تتجزأ ب الزمن لا يتجاوز 30 ثانية  
يجب أن يعمل النظام على مدار 24 ساعة.

2. (10 علامات)



- (10 علامات) .3  
U1, U2, U3, U4, U5, S, W1, W2, W3  
(10 علامات) .4



السؤال الثاني: 10 علامات

1. تحديد الممثلات (Actors)  
2. تحديد السيناريوهات (Scenarios)  
3. تحديد الحالات (Use Cases)  
4. تحديد الملاكات بين الممثلات (Relationships between actors)  
5. تحديد الملاكات بين الممثلات (Relationships between actors)  
6. تحديد كائنات التحويل الارلية (Arbitrary objects)  
7. تحديد المطلبات غير الوظيفية (Non-functional requirements)

**السؤال الثالث: (10 علامات)**

- إحدى الطرق الأساسية للعلم
- النموذج هو تمثيل مجرد لنظام يمكننا من الإجابة على أسئلة حول النظام.
- تسمح النماذج بتصور وفهم الأنظمة المعقّدة أو الواجب إيجادها.
- تفيد النماذج في فهم وتصور النظم الكبيرة والمعقدة أو البسيطة جداً

**السؤال الرابع: (20 علامة)**

- Completeness: جميع المميزات ذات الشأن تم توصيفها في المتطلبات.
- Consistency: لا يوجد متطلبين متعارضين من المتطلبات.
- Clarity: لا يمكن تقسيم متطلب بطريقتين مختلفتين.
- Correctness: تصف المتطلبات جميع المميزات للنظام والبيئة التي يقصدها الزبون والمطور.

النحو المدخل

كما ذكرنا

1- قلبة اصبعه

2- الكفاءة

3- الاعتنية

4- قلبة الاستخدام Usability ونقصد بها مدى امكانية تطبيق نظام سهلة.

و تكون الاضطرار مترددة ما بين تلك المعايير بين صعود وهبوط وبالتالي على مدى امكانية توفير تلك المعايير تكون الاضطرار من الاسباب التي ادت إلى ارتفاع سعر الـ Software وبالتالي يشهد هذه الازمة التي :

- 1- الاحتياض بحقوق الفكرية لبرامج قبل المبرمجين أو الشركات العاملة في هذا المجال .
- 2- اسعار المرتفع الذي يضعه المبرمج نفسه .
- 3- ا لكم الهائل من البرامج الموجودة في سوق العمل والتي لا يحكمها قانون .
- 4- عدم وجود مهندسي البرامج (Software Engineer) أو شرائهم في سوق العمل .

#### مواصفات الـ Software Engineer

1- شهادة متخصصة في مجال الحاسوب ( Computer Sciences ,Information Technology ) أو أي تخصص حاسوبي .

2- خبرة لا تقل عن سنة في مجال التخصص (إنشاء وتصميم الأنظمة الحاسوبية).

3- تجربة تمكنها لتطوير وتنفيذ الدائنة لكل جيد .

برمجيات الأنظمة (system software) :

خصائصها:

• تتألف من مجموعة من البرامج مكتوبة لتلبى طلبات برماج أخرى

• تعالج معلومات معقدة وتحتية في الوقت ذاته في حين أن الأنواع الأخرى تعالج بيانات غير حتمية

برمجيات الزمن الحقيقي (real-time software) :

خصائصها:

- هي البرامج التي تراقب أحداث العالم الحقيقي وتحلّلها وتحكم بها في نفس الوقت الذي تحدث فيه
- تتألف من مكون لجمع البيانات، ومكون تحليل، ومكون تحكم / إخراج ، ومكون مراقبة
- يجب أن يعطي النظام إجابة ضمن شروط زمنية محددة

برمجيات الأعمال (business software) :

خصائصها:

- هي أوسع مجال من مجالات التطبيقات البرمجية
- تطورت الأنظمة باتجاه برمجيات أنظمة المعلومات الإدارية التي تتصل بقاعدة بيانات كبيرة
- تعمل على إعادة ترتيب البيانات الموجودة بطريقة تسهل عمليات الأعمال أو اتخاذ القرارات الإدارية
- من أمثلتها: الاستعمال التفاعلي للكمبيوتر وحسابات العميل / الملف

البرمجيات العلمية والهندسية (engineering/scientific software) :

خصائصها:

- تميزت بخوارزميات سحق الأرقام

برمجيات الأجهزة (embedded software) :

خصائصها:

- انتشرت المنتجات الذكية
- تقوم بأداء وظائف خاصة ومحدودة مثل لوحة مفاتيح فرن المايكروويف أو بتوفير وظائف وإمكانات هامة مثل رقمية السيارة ولوحة أجهزة القياس

## برمجيات الكمبيوترات الشخصية (PC software) :

خصائصها:

- هي أوسع مجال من مجالات التطبيقات البرمجية
- من أمثلتها برمجيات معالجة النصوص والجداول الحاسوبية والوسائط والتسليم ..... الخ

## برمجيات الذكاء الاصطناعي (artificial intelligence software)

خصائصها:

- تعتمد على خوارزميات غير رقمية لحل مسائل معقدة لا يمكن ردتها إلى الحساب أو التحليل المباشر
- من أنواعها الأنظمة الخبرية وتسمى أنظمة قواعد المعرفة
- من فروعها الشبكات العصبية الصناعية

الخصوصية Confidentiality  
الخصوصية Confidentiality: خصوصية العملاء الذين نتعامل معهم حيث يجب علينا نحن

المهندسين الحفاظ على أسرارهم ، فعند تصميم النظام يجب علينا مراعاة ذلك .

الشخصية Competence : لا تضع نفسك في مكان ليس مكانك فيجب عليك تحديد للعميل  
شخصك .

الحفظ على حقوق الملكية .

استخدام الحاسوب Computer Misuse لا يستخدم الحاسوب إلا في الشيء الذي أعد من أجله

- اولا الاعتمادية تحدثنا في هنا مسبقا وهو مدى ثقة المنظمة بالنظام اما الاشياء التي تعمل على رفع الاعتمادية ف هي :
- **Hardware Reliability** فلو كان لدينا نظام ذات جودة عالية لكن المعدات كانت رديئة فذلك يؤدي إلى ضعف الاعتمادية.
- **Software Reliability** وبالمثل لو أن المعدات عالية المستوى ومن ثم يكون النظام رديء أو أن تكون قاعدة البيانات الخاصة بالنظام ضعيفة الحماية فذلك يؤدي إلى ضعف الاعتمادية في النظام.
- **Operator Reliability** فلو كان المستخدم لا يجيد استخدام النظام فقد يقوم بعمل اشياء تضر بالمنظمة ككل.

الآن لراجحه  
وتستخدم هذه الطريقة اذا كانت متطلبات النظام واضحة المعالم ومحددة وذلك لأن هذه الطريقة اذا انتقلنا من احد المراحل فلا عودة

بعدها كما هو موضح بالرسم وذلك لأنها ستؤدي الى خسائر فعلى سبيل المثال لو اننا حددنا المعدات المفترض تواجدها وعلى

هذا الاساس قام المبرمج بابرام العقد مع الشركة الراغبة في النظام وعلى اساس المعدات والتكليف التي اعدها لهم المندس او المحلل ومن ثم لاحظ ان المعدات ناقصة فان تكاليف المعدات الناقصة سوف يتحملها المهندس

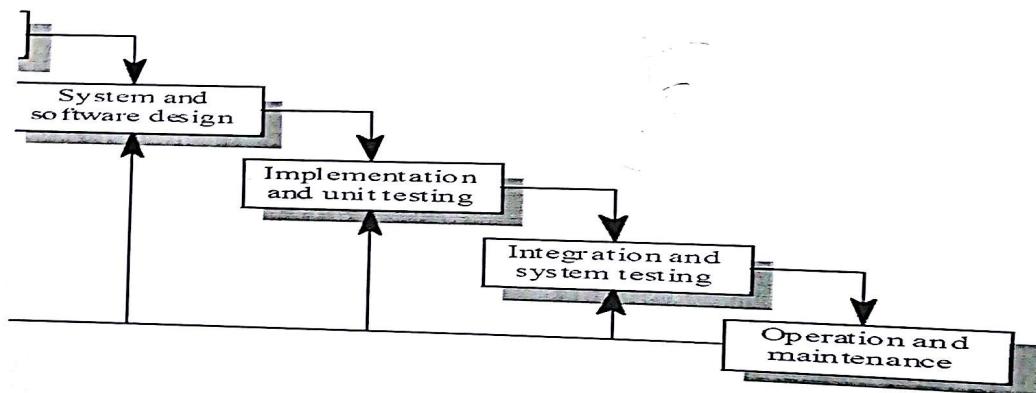
■ تعريف وتحليل وتحديد المتطلبات. Requirements analysis and definition.

■ تصميم النظام وتصميم البرمجيات. System and software design.

■ تفخيد واختبار وحدات النظام. Implementation and unit testing.

■ تجميع النظام وختباره. Integration and system testing.

■ عمل النظم وصيانته. Operation and maintenance.



# سلع رَصْبِحَ مَادَهْ هَنْدَرْتَهْ جَمِيعَهْ سَالَهْ

## دررة اقتصاد ١٦ ٢٠١٧ـ المُالُ الْأَدَلُ (٢٠ درعَه)

من خلال ما سبق يمكننا أن نحدد النقاط التي يجب على المستخدم أن يقوم ببراعتها قبل وبعد واتاء تصميمه لأحد هنا التوعين من البرامجيات .

(3)

١- الزمن Time : وتفصيل بالزمن من ثلاث نقاط اسلية :

أـ. تحديد موعد تسليم النظام .

بـ. تحديد الفترة الزمنية للمبرمجين لاتخاذ هذا النظم .

تـ. تحديد سرعة النظم أو ما يسمى استغلال موارد النظم .

٢- الجودة Quality : ويمكننا أن ننظر إلى الجودة من ثلاث وجهات نظر :

أـ. العميل Customer : بالنسبة إلى العميل فله يهتم من ناحية الجودة التالية:

• تسليم النظم في الموعد المحدد .

• تحقيق الاعتمادية والأمانة والامان في النظم Security & Reliability (Safety)

وتفصيل بالاعتمادية تقييد الاعمال من دون اخطاء .

• الكفاءة (تفاوت اكبر قدر ممكن من العمليات في نفس وقت ) .

• قابلية الصيانة Maintainability (المرونة الكافية للتغيير في العلبات أو اضفافها او تغيير الصلاحيات والمستخدمين .....).

بـ. المستخدم User : ينظر المستخدم في جودة من النظم من حيث التالي :

• أن يكون من وسهك التعلم جيد التصميم .

• الاعتمادية .

• الكفاءة .

تـ. المطور Developer Or Software Engineer : بالنسبة للجودة في نظر مهندس البرمجيات فهو ينظر لها من وجهات التالية :

• عمل الامن والامان يكون عليا جدا .

• جودة التصميم الخارجي Design .

• الاعتمادية .

• الكفاءة .

• قابلية الصيانة .

# الحوال إلى (حاجة)

الميادنة : يجب أن تتضمن البرامج لتنمية الاحتياجات المتغيرة

الثقة: يجب أن تكون برامج جديدة بالكلة

المكانة: يجب عدم المهر في استخدام موارد النظام

المقاولة: يجب أن تقبل البرنامج من قبل المستخدمين الذي صممت من أجلها. وهذا يعني أنه يجب أن تكون مفهومة وقابلة للاستخدام ومتواافق مع أنظمة أخرى.

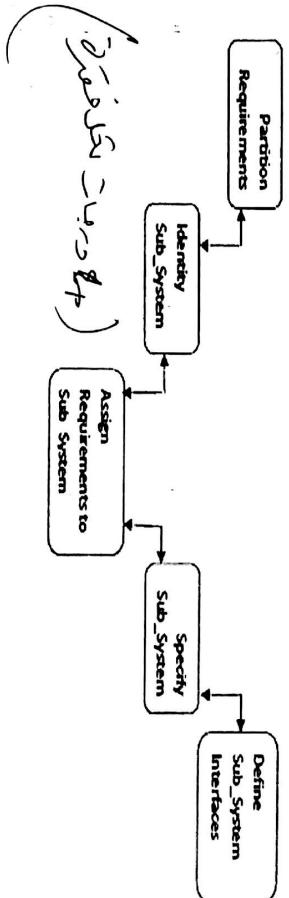
عد التجانس ، والتسليم والثقة.

عد التجانس: تطوير تقنيات لبناء البرمجيات التي يمكن التعامل مع منصات غير متحاشية وبيئات التنفيذ

التسليم: تطوير التقنيات التي تؤدي إلى سرعة تسلیم البرمجيات

الثقة: تطوير التقنيات التي يمكن الوثوق بها بثبت هذا البرنامج من قبل مستخدميها.

الحالات



**Partition Requirements – 1 ■**

تقسيم المتطلبات وكذلك تحديد متطلبات كل قسم من أقسام النظام سواء كانت تلك المتطلبات هي معدات كالاطارات أو كاميرات المراقبة أو كانت برمجيات

معينة فمثلا لو كان النظام كبير الحجم فان الشركة المصنعة له سوف تقسم النظام وتعطي كل فريق عمل من المبرمجين او المحللين جزء من النظام الكلي.

**Identity Sub\_System - 2** ■  
تعريف الانظمة الفرعية وذلك حسب تقسيمنا للمتطلبات.

**Assign Requirements to Sub\_System - 3** ■  
تأثير تلك المتطلبات المحددة على الانظمة الفرعية بمعنى هل تلك المتطلبات تواءم الانظمة الفرعية اذا كان نعم فتنقل الى المرحلة التالية واذا كان لا فنعود لنحدد ونقسم المتطلبات والانظمة الفرعية لاحظ اننا وضعنا سهم ذات جهتين حيث بامكاننا ان نعود ان اردنا ذلك .

**Specify Sub\_System - 4** ■  
نحدد ونعطي مواصفات لكل نظام فرعى ان اننا سوف نحدد وظيفة كل جزء من اجزاء

**Define Sub\_System Interfaces - 5** ■  
النظام الفرعى فلا يكون هناك نظام فرعى دون عمل مثلا.

تحديد الشكل والواجهات الخاصة بالنظام.

## الحوال الرابع (٢٥ درج)

وتستخدم هذه الطريقة اذا كانت متطلبات النظام واضحة المعالم ومحددة وذلك لأن هذه الطريقة اذا انتقلنا من احد المراحل فلا عودة بعدها كما هو موضح بالرسم وذلك لأنها ستؤدي الى خسائر فعلى سبيل المثال لو اتنا حدثنا المعدات المفترض تواجدها وعلى هذا الاساس قام المبرمج بابرام العقد مع الشركة الراغبة في النظام وعلى اساس المعدات والتكاليف التي اعدها لهم المهندس او المحلل ومن ثم لاحظ ان المعدات ناقصة فان تكاليف المعدات الناقصة سوف يتحملها المهندس

■ تعريف وتحليل وتحديد المتطلبات.

■ تصميم النظام وتصميم البرمجيات. System and software design.

■ تنفيذ وإختبار وحدات النظام. Implementation and unit testing.

(O) ■ تجميع النظام وإختباره. Integration and system testing.

عمل النظم وصيانته. Operation and maintenance

■ من عيوب نموذج الشلال صعوبة تقدير وتكييف التغييرات أثناء العملية.

■ من مشاكل نموذج الشلال: التقسيم غير المرن للمشروع إلى مراحل منفصلة، يزيد من صعوبة الإستجابة عن تغيير متطلبات المستهلك، لهذا يصبح هذا النموذج مفيداً فقط ومرغوباً عند التفهم الكامل للمتطلبات وقلة التغييرات فيها إلى الحد الأدنى.

