Problem Solving Techniques
تقنيـــات حلـــول المشـــاكل

المحـــاضرة الأولـــى

ربيع 2023

1

بنـهاية الفصل الدراسي يكون الطالب قادراً على أن:

- ✓ يُعرِّف المشكلة Problem Definition
- ✓ يُحدد مراحل حل مشكلة Problem Solving Stages
- ✓ يكتب خطوات حل المشكلة الخوارزمية "Algorithm"
- ✓ يرسم خرائط التدفق Flow chart
- ✓ يكتب برنامج متكامل بلغة بايثون Python
- ✓ يختبر البرنامج ويصحح اخطائه.

2

يمكن تعريف المشكلة/ المشاكل Problem(s)

- ✓ لغويا: هي كلمة تستعمل للتعبير عن خلل ما كبيرًا كان أم صغيرًا وعلى كافة مستويات الأهمية.
- ✓ تقنيا: هي حالة عدم القدرة على بلوغ هدف Object ما.
- ✓ رياضيا: الهدف + المعوق = المشكلة

Object + Obstacle = Problem

الهدف Object: هو الهدف الذي نرغب في تحقيقه أو الناتج المطلوب الوصول إليه

المعوق Obstacle: هو ما يمنعنا من تحقيق الهدف أو يعيق الوصول إليه.

3

علم حل المشاكل (Problem Solving)

- ➢ هو علم كيفية التغلب على العوائق لتحقيق الأهداف، وهو علم تعلـــم التفكير بشكل منطقي Logical تسلسلي Sequential. أي الوصول إلى هدف أو ناتج محدد مطلوب من خلال خطوات وأنشطة متتابعة ومعطيات محددة.

4

علم حل المشاكل (Problem Solving)

- كل مشكلة فريدة من نوعها من ناحية (المعلومات الخاصة بها Information وعمليات معالجتها Solving Process) إلا أنها جميعًا يتبع فيها ذات مراحل الحل Solution Stages وفق نسق ثابت.

ملاحظة: من المهم جدًا تنفيذ خطوات الحل بشكل منطقي متسلسل مع الانتباه إلى إن الخطأ في تنفيذ أي خطوة قبل أخرى **قد** يتسبب في عدم الحصول على الحل الصحيح وبالتالي نتائج غير مرغوب فيها.

5

مراحل حل المشكلة (problem solving stage)

أولا :يُعرِّف المشكلة وتحليليها Problem Definition & Analysis

تحليل عناصر المشكلة Problem Analysis:

أي تحديد المخرجات المطلوبة والمدخلات المتوفرة، وعمليات المعالجة الحسابية أو المنطقية. وفي هذه المرحلة يتم تحديد أبعاد المسألة والهدف المطلوب الوصول إليه عن طريق الاتي:

- تعريف المخرجات وشكلها بدقة (النتائج المراد تحقيقها من المسألة).
- تحديد المدخلات بناءً على المخرجات المطلوبة.
- تحديد طرق الحل المختلفة وتقييمها لاختيار أفضلها من حيث السهولة وسرعة التنفيذ والمساحة التي تحتاجها من الذاكرة.

6

مراحل حل المشكلة (problem solving stage)

ثانيًا :تصميم منطق الحل Solution design: في هذه الخطوة يتم وصف الحل في صيغة قابلة للتنفيذ ويتمثل في:

- **الخوارزميات (Algorithms).**
- **المخططات الانسيابية (flowcharts).**
- **جداول القرار (Decision tables).**

بعد تعريف وتحليل المشكلة من مخرجات ومدخلات التي **يتم إعداد خطة الحل** يجب أن تكون على شكل سلسلة من الخطوات **المتتالية** ويطلق على هذه الخطوات (الخوارزمية Algorithm) نسبة لعالم الرياضيات ومؤسس علم الجبر محمد بن موسى الخوارزمي، وتمثيل خطوات الحل من خلال خرائط التدفق (Flowchart).

مراحل حل المشكلة (problem solving stage)

ثالثًا :تصميم البرنامج على الكمبيوتر Program Design

بعد الانتهاء من عمل خريطة التدفق (Flowchart) ولحل المشكلة باستخدام الحاسوب نقوم بترجمتها إلى إحدى لغات البرمجة.

رابعاً: اختبار صحة البرنامج وتصحيح أخطائه Program Testing

أثناء كتابة البرنامج نقع في بعض الأخطاء بدون قصد. قد تكون لغوية نتيجة لمخالفة قواعد لغة البرمجة ويمكن اكتشافها وتصحيحها بسهولة، أو أخطاء أثناء الكتابة البرنامج بحيث نكتب علامة الطرح بدل الجمع مثلا مع عدم مخالفة القواعد، وبالتالي لا نستطيع اكتشاف هذه الأخطاء إلا عندما إدخال بيانات نعرف نتائجها مسبقا، **ولكي نتمكن من اكتشاف الأخطاء وتصحيحها لابد من معرفة أنواع الأخطاء**

يمكن تصنيف الأخطاء إلى3 أنواع:

- **خطأ لغوي (Syntax Error)**: يحدث نتيجة لمخالفة قواعد لغة البرمجة وعادة ما يتم اكتشافه من قبل البرنامج المترجم.
- **خطأ منطقي (Semantic Error)**: وهو من الأخطاء التي لا يكتشفه البرنامج المترجم، وعادة ما يتم اكتشافه من قبل المبرمج بتجربة البرنامج ببعض البيانات التي يتوقع نتائجها. بحيث يكون البرنامج صحيح لغويًا ويُنفذ ولكن يُعطي نتائج غيرالمطلوبة.
- **خطأ زمن التنفيذ (Runtime Error)**: ويسمى أحيانا بـExceptions. وهذا النوع من الأخطاء يحدث نتيجة حدوث خطأ أثناء تنفيذ جمل البرنامج مثلا : القسمة على صفر والفيض (OverFlow, Under Flow)

9

مراحل حل المشكلة (problem solving stage)

خامسًا :توثيق البرنامج Program Documentation

يتم كتابة كل الخطوات التي اتخذت لحل المشكلة من مدخلات، ومخرجات، وخطة حل، وخريطة التدفق المستخدمة، واللغة التي كُتب بها البرنامج، أوامر البرنامج وتاريخ آخر تعديل للبرنامج، ومن شارك في عمل البرنامج؛ للاحتفاظ به موثق للرجوع إليه في أي وقت بهدف التصحيح أو التطوير.

وهذا يفيد في حال اشتراك أكثر من شخص في كتابة البرنامج أو عند التعديل في البرنامج بواسطة أشخاص آخرين.

10

خطوات حل أي مشكلة بواسطة الحاسوب

مما شُرح سلفاً نستنتج ان خطوات حل أي مشكلة بواسطة الكمبيوتر هي:

- ✓ تعريف المشكلة او المسألة وتحليلها Problem Definitions and Analysis
- ✓ وضع خوارزمية الحل (Algorithm) والمخطط الانسيابي (Flowchart).
- ✓ كتابة البرنامج بإحدى لغات البرمجة (Writing the program) وترجمة البرنامج الي لغة الآلة (Compilation)
- ✓ تنفيذ البرنامج (Execution)

11

استخدام الخوارزميات في حل المشكلات:

الخوارزمية: هي مجموعة من الخطوات التي نصل إلى الحل المطلوب عند تنفيذها أي وصف منطق الحل في خطوات:

منفصلة، واضحة، مرتبة، متكاملة، وصحيحة

باستخدام لغة تعبير مناسبة لتحقيق الخصائص السابقة، بحيث تمثل حلا عاما يحقق الهدف ويعطي النتائج المطلوبة.

لكتابة الخوارزمية يجب:

1) تحديد المشكلة.
2) تحديد المخرجات.
3) تحديد المدخلات.
4) اهمال البيانات غير المفيدة في حل المشكلة.
5) تحديد خطوات الحل (العمليات).
6) كتابة الحل Algorithm.

12

خواص الخوارزميات Algorithms :

- ✓ لها بداية ونهاية.
- ✓ تكتب الخوارزمية بصفة عامة بحيث تكون صالحة لجميع الاحتمالات التي قد تطرأ في معطيات المسألة.
- ✓ التعليمات الواردة بالخوارزمية يجب ان تكون محددة المعني، وغير قابلة لتعدد التأويل.
- ✓ أن تتوقف العمليات بعد عدد محدد من الخطوات.
- ✓ أن تؤدي العمليات إلى الحل الصحيح للمسألة .

13

الشفرة Pseudo Code المستخدمة في كتابة الخوارزميات:

1) تعليمة البداية (ابداء Start او Begin) والنهاية (توقف Stop أو End)
2) تعليمة الادخال (إقراء Read) او (أدخل Input).
3) تعليمة الاخراج (اكتب Write) او (اطبع Print).
4) تعليمة حسابية مثل (أضف add (i=i+1) او اطرح Sub او...).
5) تعليمة شرطية، وهي العليمة التي يشترط في تنفيذها تحقق شرط منطقي

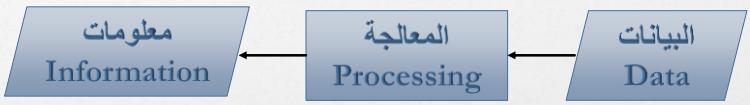
مثل: اذا كان س>0 اطبع «موجب»

```
if x>0 print "positive"
```

6) حلقات التكرار Loops

14

تذكر دائما



مدخلات
Input

التنفيذ

مخرجات
Output

15

التمرين رقم 1: طريقة غرس شتلة الزيتون

الوصف : شتول الزيتون تأتي في أشكـال متنوعة: شتلة بذور، شتلة عقلة، شتلة "غصن"، و بناءًا عليه يتم غرس هذه الأنواع بطرق مختلفة يراعى فيها الأسس العلمية لعملية الغرس.

المشكلة : كيف يمكن غرس شتلة زيتون بصورة صحيحة .؟؟؟

الهدف : تحديد كيفية غرس شتلة الزيتون.

النطاق : هذه المسألة تتعلق فقط بشتول البذرة.

القيود : الغرس حسب القواعد العلمية المتعارف عليها.

المدخلات : ـ شتلة الزيتون ـ موقع الغرس- أدوات الغرس

المعالجة : الغرس حسب الخطوات العلمية المتبعة في غرس شتول الزيتون

المخرجات : شتلة زيتون مغروسة في الموقع الطبيعي .

16

خوارزمية غرس شتلة الزيتون .

- إبدا .
- تحديد الموقع المناسب للغرس.
- تجهيز الحفرة.
- نزع الشتلة من الكيس.
- وضع الشتلة في مركز الحفرة.
- ردم التراب حول جذر الشتلة.
- ضغط التراب لتثبيت الشتلة.
- تجهيز حوض حول جذع الشتلة.
- ري الشتلة بالماء .
- النهاية .

- تقدير عمق الحفرة مبدئيا.
- حفر الحفرة.
- قياس عمق الحفرة باستخدام كيس الشتلة .
- تسوية الحفرة .

17

استخدام المخطط الانسيابي Flow Chart في حل المشكلات:

المخطط الانسيابي: هو عبارة عن تمثيل تخطيطي باشكال هندسية لغرض توضيح الخوارزمية، يوضح بسهولة ترتيب خطوات حل المشكلة، بدءا من إدخال البيانات، ثم تحديد العمليات الحسابية والمنطقية، وصولا للمخرجات التي تمثل حل المشكلة.

- اي ان المخطط الانسيابي هو عبارة عن مجموعة من الاشكال الهندسية، كل شكل يذل على تعليمة معينة، تتصل الاشكال بأسهم تبين اتجاه التنفيذ، وتكتب التعليمة داخل الشكل المناسب لها.

الرموز والأشكال الهندسية المستخدمة في رسم مخططات الانسياب

الرقم	الشكل	الاستخدام	توضيح (مثال)
1		يستخدم في تمثيل بداية البرنامج ونهايته	ابدأ / توقف
2		يستخدم في تمثيل إدخال البيانات أو اخراج النتائج (طباعتها)	ادخل قيمة س / اطبع قيمة س
3		يستخدم في تمثيل معالجة البيانات مثل العمليات الحسابية	أجعل س=ص+2
4		يستخدم في تمثيل اتخاذ قرار معين من خيارات (مثل المقارنات)	هل س= 0 ؟ نعم / لا
5		تستخدم خطوط الانسياب(الاسهم) في تمثيل الانسياب المنطقي للبرنامج	هل س>0 6 ؟ نعم / لا
6		تستخدم في توصيل الاجرآت المختلفة في المخطط أو لبدء صفحة جديدة في حالة عدم الاتساع	

الاعتبارات الواجب مراعاتها عند رسم المخطط الانسيابي:

- ✓ بساطة ووضوح المخطط لسهولة تتبع خطواته.
- ✓ الاتجاه الافتراضي لأشكال المخطط من اليسار إلى اليمين ومن أعلى إلى أسفل.
- ✓ رمز العمليات يخرج منه خط اتجاه واحد فقط.
- ✓ شكل اتخاذ القرار يدخل له خط اتجاه واحد ويخرج منه خطان (نعم/لا).
- ✓ الرمز الطرفي للبداية له خط اتجاه واحد خارج والعكس في الرمز الطرفي للنهاية.
- ✓ يفضل استخدام رمز اضافة تعليق مع أشكال الخريطة التي تحتاج توضيح.
- ✓ يستخدم رمز الربط أو الاتصال إذا كانت خريطة التدفق كبيرة وتحتاج أكثر من صفحة.
- ✓ بعد رسم خريطة التدفق يفضل تتبع جميع مساراتها واختبارها بقيم افتراضية معروف نتائجها.

20

مميزات استخدام المخططات الانسيابية:

- ✓ تمثل ضرورة قبل كتابة البرامج الكبيرة.
- ✓ تمثل أحد أشكال توثيق البرنامج.
- ✓ تضع تصورا كاملا لحل المشكلة وتساعد في تتبع مسارها.
- ✓ تساعد في عدم تكرار أجزاء معينه في الرسم أو البرمجة.
- ✓ الأشكال المستخدمة في رسم خرائط التدفق لها مدلول واحد لدى جميع المتعاملين معها.
- ✓ تساعد في تصحيح الأخطاء بسهولة.
- ✓ تساعد في تطوير وصيانة البرامج.

21

عيوب استخدام خرائط التدفق:

- × قد تبدو الخريطة معقدة للمشكلات الكبيرة.
- × بعض التعديلات في البرنامج قد يؤدي لإعادة رسم الخريطة.
- × أحيانا تشكل نسخ خريطة التدفق صعوبة كبيرة.
- × الوقوع في بعض التفاصيل التي تبعدنا عن الحل.

22

صياغة حل المسائل

المسألة1 : قراءة عددين وايجاد حاصل جمعهما؟

أولاً : تحليل عناصر المسألة : لتحليل عناصر المسألة نقوم بما يلي:

المعطيات المطلوب

المعطيات: العدد الاول (س) ، العدد الثاني (ص)
المطلوب : حساب المجموع (م) ،
المعالجة : م = س + ص

23

الخوارزمية

ثانياً : تصميم منطق الحل Solution design:

كتابة الخوارزمية بلغة تعبير مناسبة

لك الخيار أن تكتب الخوارزمية باللغة العربية أو الانجليزية

1) Start
2) Read first (F), second (S)
3) Sum (S)=F + S
4) print Sum
5) Stop.

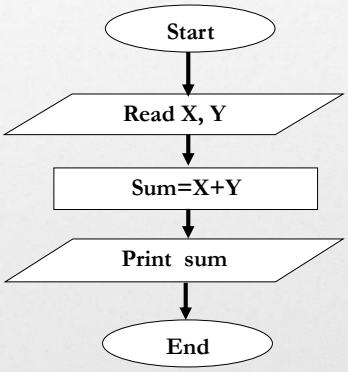
1) البداية
2) اقرا العدد الاول (س)
3) اقرا العدد الثاني (ص)
4) احسب م= س + ص
5) اطبع (م)
6) النهاية

24

المخطط الانسيابي

ثانياً: تصميم منطق الحل Solution design:

رسم المخطط الانسيابي للخوارزمية



25

ثالثاً: تصميم البرنامج على الكمبيوتر Program Design

أي كتابة البرنامج بإحدى لغات البرمجة: البرنامج بلغة بايثون

```
1 x=int (input ("x= "))
2 y=int (input ("y= "))
3 sum=x+y
4 print("sum= " ,sum)
```

26

رابعاً: اختبار صحة البرنامج وتصحيح أخطائه Program Testing

اختبار البرنامج وتتبعه: أي تتبع البرنامج باعطائه قيم ومقارنة الناتج بما هو متوقع مثلا:

- القيمة الأولى: 20 ،، القيمة الثانية: 30

الناتج لابد ان يكون 50 وإلا فهناك خطأ

```
1 x=int (input ("x= "))
2 y=int (input ("y= "))
3 sum=x+y
4 print("sum= " ,sum)
```

البرنامج

```
Python3IDE(Python 3.7) running!
x= 20
y= 30
sum=  50
```

ناتج تنفيذ البرنامج

نهاية المحاضرة الاولى

27

صياغة حل المسائل

المسألة2: حساب مساحة دائرة بمعلومية نصف قطرها

أولاً : تحليل عناصر المسألة : لتحليل عناصر المسألة نقوم بما يلي:

المعطيات — المطلوب

المعطيات: نصف قطر الدائرة (نق) أو r

المطلوب : حساب مساحة الدائرة (م) أو Area

المعالجة : مساحة الدائرة م = ط × نق^2

$$Area = \pi r^2$$

28

الخوارزمية

ثانياً : تصميم منطق الحل Solution design:

كتابة الخوارزمية بلغة تعبير مناسبة

لك الخيار أن تكتب الخوارزمية باللغة العربية أو الانجليزية

1) Start
2) Read r
3) let Pi=3.14
4) $Area = Pi * r^2$
5) Print $Area$
6) Stop.

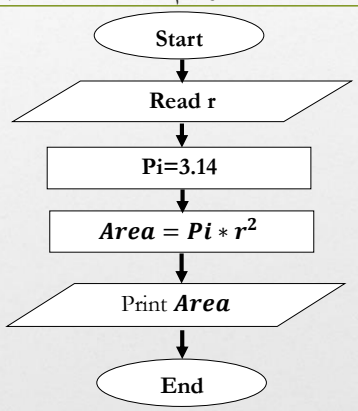
1) البداية
2) اقرا نصف القطر (نق)
3) دع ط = 3.14
4) احسب م= ط × نق^2
5) اطبع (م)
6) النهاية

29

المخطط الانسيابي

ثانياً: تصميم منطق الحل Solution design:

رسم المخطط الانسيابي للخوارزمية



30

ثالثاً: تصميم البرنامج على الكمبيوتر Program Design

أي كتابة البرنامج بإحدى لغات البرمجة: البرنامج بلغة بايثون

```
1 r=float(input("r= "))
2 pi=3.14
3 area=pi*r**2
4 print("Area= " , area)
5
```

31

رابعاً: اختبار صحة البرنامج وتصحيح أخطائه Program Testing

اختبار البرنامج وتتبعه: أي تتبع البرنامج باعطائه قيم ومقارنة الناتج بما هو متوقع مثلا:

- اعطي قيمة لنصف القطر

```
1 r=float(input("r= "))
2 pi=3.14
3 area=pi*r**2
4 print("Area= " , area)
```

البرنامج

```
Python3IDE(Python 3.7) running!
r= 3.5
Area=  38.465
```

ناتج تنفيذ البرنامج

نهاية المحاضرة الاولى

32