



دليل المعلم

10.1

الصف العاشر ■ الفصل الدراسي الأول ■ الجزء الأول



دليل المعلم

لتقنية المعلومات

10.1

الصف العاشر

إشراف

أ/ منى سالم عوض سالم

أ/ فاطمه خليف غنيم العنزلي

أ/ ماجدة إبراهيم مزعل الدوسري

أ/ ساره ياسين عبدالله الامير

أ/ إيمان عبد العزيز أحمد الفارسي

لجنة المواءمة

أ/ منار مصطفى عبد الحميد جمال

أ/ منى مرزوق مخلد العازمي

أ/ إبراهيم عبدالله إبراهيم المياس

إخراج

أ/ سنيه محمد علي المؤمن

6	مقدمة
7	1 الاستراتيجية التعليمية
18	2 الوحدة الأولى: الأمن السيبراني Cyber Security
19	- وصف الوحدة : الأمن السيبراني Cyber Security
20	- خطة الدروس : الأمن السيبراني Cyber Security
21	- نواتج التعلم : الأمن السيبراني Cyber Security
21	- مصطلحات حاسوبية : الأمن السيبراني Cyber Security
24	- استراتيجيات تعليمية مقترحة : الأمن السيبراني Cyber Security
24	- أوراق العمل : الأمن السيبراني Cyber Security
25	3 الوحدة الثانية: برمجة بايثون Python
26	وصف الوحدة : برمجة بايثون Python
27	الدرس الأول : مدخل إلى البرمجة Introduction to programming
27	- خطة الدروس: مدخل إلى البرمجة
28	- نواتج التعلم : مدخل إلى البرمجة
28	- مصطلحات حاسوبية: مدخل إلى البرمجة
31	- ملاحظات للمعلم : مدخل إلى البرمجة
34	- أوراق العمل: مدخل إلى البرمجة
37	الدرس الثاني : المتغيرات Variables
37	- خطة الدروس: المتغيرات Variables
38	- نواتج التعلم: المتغيرات Variables
38	- مصطلحات حاسوبية: المتغيرات Variables
39	- استراتيجيات تعليمية مقترحة : المتغيرات Variables
40	- أوراق العمل : المتغيرات Variables
44	الدرس الثالث : السلاسل النصية Strings
45	- خطة الدروس: السلاسل النصية Strings
46	- نواتج التعلم: السلاسل النصية Strings
46	- مصطلحات حاسوبية: السلاسل النصية Strings
47	- استراتيجيات تعليمية مقترحة : السلاسل النصية Strings
48	- أوراق العمل: السلاسل النصية Strings
55	الدرس الرابع : الشروط Conditions
55	- خطة الدروس: الشروط Conditions
56	- نواتج التعلم: الشروط Conditions
56	- مصطلحات حاسوبية: الشروط Conditions

57	- استراتيجيات تعليمية مقترحة : الشروط Conditions
57	- أوراق العمل: الشروط Conditions
66	الدرس الخامس : التكرار Loop
66	- خطة الدروس: التكرار Loop
67	- نواتج التعلم: التكرار Loop
67	- مصطلحات حاسوبية: التكرار Loop
69	- استراتيجيات تعليمية مقترحة : التكرار Loop
71	- أوراق العمل: التكرار Loop
75	الدرس السادس : تصحيح الأخطاء والاستثناءات Debugging and Exceptions
76	- خطة الدروس: تصحيح الأخطاء والاستثناءات
76	- نواتج التعلم: تصحيح الأخطاء والاستثناءات
77	- مصطلحات حاسوبية: تصحيح الأخطاء والاستثناءات
78	- ملاحظات للمعلم : تصحيح الأخطاء والاستثناءات
80	- أوراق العمل : تصحيح الأخطاء والاستثناءات
86	4 الوحدة الثالثة: المنتجات الرقمية
87	- المنتجات الرقمية : المشروع
87	- نواتج التعلم: المنتجات الرقمية
88	- مراحل إعداد المشروع
90	- المهارات المقترحة للمشروع ومتابعة إعدادة
92	- أمور يجب مراعاتها في وحدة المنتجات الرقمية
93	5 تحضير الدروس
97	- خطوات إعداد الدرس
98	- نموذج تحضير درس مادة الحاسوب للصف العاشر
101	6 التقويم التربوي
103	- استراتيجيات التقويم
106	- توزيع درجات مادة الحاسوب
107	7 الاختبارات
108	- الاختبارات التحصيلية
109	- الميزان النسبي/جدول مواصفات الاختبار
111	- ميزان نسبي مقترح
113	الخاتمة

المقدمة

عزيزي المعلم

يسعدنا أن نقدم لك هذا الدليل الشامل الذي يهدف إلى دعمك في تقديم مادة تقنية المعلومات لمتعلمي الصف العاشر بما في ذلك مفاهيم الأمن السيبراني والبرمجة بلغة بايثون بصورة فعالة ومبتكرة.

يتضمن هذا الدليل استراتيجيات تعليمية متنوعة تتناسب مع الفئة العمرية للمتعلمين، مثل التعلم المباشر (المحاضرة)، التعلم التعاوني، المناقشة والحوار، التعلم القائم على حل المشكلات، والتعلم القائم على المشاريع ولعب الأدوار، كما يوفر الدليل أنواعًا مختلفة من التقويمات، سواء التكوينية منها أو النهائية، لضمان تقييم شامل لتقدم المتعلمين ومساعدتهم على تحسين أدائهم بشكل مستمر.

لضمان تجربة تعليمية شاملة، يحتوي هذا الدليل على خطط مفصلة لكل وحدة دراسية تشمل أهداف الوحدة، الموضوعات الرئيسية، الأنشطة التعليمية، المواد والأدوات اللازمة، وأساليب التقويم. وبعض الاستراتيجيات المقترحة لتناول محتوى محدد.

فيما يخص الأمن السيبراني، ستجد في هذا الدليل توجيهات حول كيفية تقديم مفاهيم الأمن الأساسية، مثل الحفاظ على سرية البيانات وسلامتها وإمكانية الوصول إليها. ستتعلم كيفية تعريف المتعلمين على التهديدات السيبرانية المختلفة وطرق الوقاية منها.

أما بالنسبة للبرمجة بلغة بايثون، فإن الدليل يحتوي على موارد مفيدة لشرح المفاهيم البرمجية الأساسية، من خلال أمثلة وتطبيقات إضافية، استراتيجيات مقترحة، أدوات تقييم مختلفة قد تساعدك عزيزي المعلم في الحصة الدراسية.

كما يتضمن الدليل قائمة شاملة بالمصطلحات الحاسوبية الهامة، بالإضافة إلى الحلول النموذجية لأوراق العمل لتيسير عملية التصحيح. يهدف هذا الدليل إلى تزويدك بالأدوات والمعارف اللازمة لتحسين عملية التعلم والتعليم، ودعمك في التحضير الفعال للدرس وتنفيذه بنجاح.

نحن على يقين بأن هذا الدليل سيكون مرشدك الأمثل في رحلتك التعليمية، ويساعدك على تقديم دروس ممتعة، تساهم في تطوير مهارات المتعلمين وتعزيز فهمهم للتكنولوجيا، الأمن السيبراني، والبرمجة بلغة بايثون. نأمل أن تجد في هذا الدليل كل ما تحتاجه لتحقيق أفضل النتائج في تدريس منهج الصف العاشر.

مع أطيب التمنيات

01

الاستراتيجيات التعليمية



الاستراتيجيات التعليمية:

عزيزي المعلم

في ظل التطور السريع في مجال التعليم وتكنولوجيا المعلومات، أصبح من الضروري اعتماد إستراتيجيات تعليمية مبتكرة وحديثة ومصادر تعلم متنوعة لتلبية احتياجات المتعلمين وتحفيزهم على التعلم الفعّال، مثل التعلم التعاوني والتعلم القائم على حل المشكلات، بالإضافة إلى التعلم القائم على المشاريع والعصف الذهني وغيرها من الإستراتيجيات التي تتناسب مع مادة تقنية المعلومات والفئة العمرية للصف العاشر، والتي بدورها تساعد المتعلمين على تطوير مهارات التفكير النقدي والإبداعي، وتوسع من مدارك المتعلمين وتعزز فهمهم.

ومن المصادر التي تعتمد عليها هذه الإستراتيجيات : الكتب الدراسية والمراجع الإلكترونية والفيديوهات التعليمية وغيرها من المصادر الحديثة، بالإضافة إلى ذلك يلعب التقويم دورًا حيويًا في العملية التعليمية، مثل التقويم التشخيصي والتكويني والختامي، وإستراتيجياتها المختلفة، وأدواتها التي يجب أن يستخدمها المعلمون لتقييم تقدم المتعلمين، وتنوع هذه الأدوات لتشمل أساليب متنوعة تلبي احتياجات المتعلمين وتوفر تغذية راجعة شاملة من اختبارات عملية ومشاريع وغيرها، حيث تلعب دورًا أساسيًا في قياس تقدم المتعلمين وتحديد نقاط القوة والضعف لديهم، وبتكامل هذه العناصر يمكننا تحقيق بيئة تعليمية شاملة تشجع على التعلم المستمر وتدعم تحقيق الأهداف التعليمية.



الاستراتيجيات التعليمية:

الاستراتيجيات التعليمية هي الخطط والأساليب والتقنيات التي يستخدمها المعلمون لتسهيل وتحسين عملية التعلم والتعليم، تهدف هذه الإستراتيجيات إلى جعل التعليم أكثر فعالية وكفاءة وجاذبية للمتعلم، يمكن أن تتضمن الإستراتيجيات التعليمية مجموعة متنوعة من الأنشطة والتدريبات والتوجيهات التي تساعد المتعلمين على فهم المفاهيم واكتساب المهارات والمعرفة بطرق متنوعة تلائم احتياجاتهم التعليمية.

استراتيجية التعلم المباشر (المحاضرة) :



التعلم المباشر هو أسلوب تعليمي يعتمد على تقديم المعلم للمعلومات مباشرة إلى المتعلمين في بيئة صفية تقليدية، يكون المعلم هو المصدر الأساسي للمعرفة، بينما يكون دور المتعلم هو الاستماع والتدوين وطرح الأسئلة، والمشاركة في النقاشات، يكون أكثر فاعلية عند تدريس فكرة أو مفهوم.

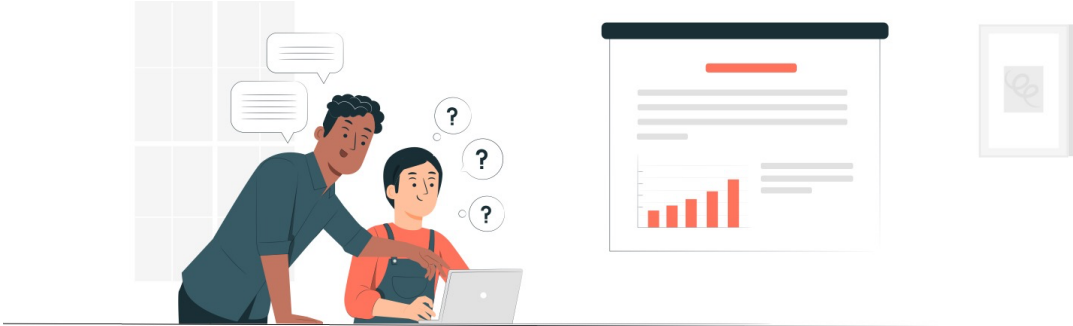
تقديم المعلم أمثلة لبعض الوظائف المستحدثة في مجال الأمن السيبراني.
شرح مناطق نافذة محرر برنامج PyCharm.

مثال:



الاستراتيجيات التعليمية:

استراتيجية المناقشة والحوار :



إستراتيجية المناقشة والحوار هي أسلوب تعليمي يهدف إلى تعزيز التعلم من خلال تبادل الأفكار والمعلومات بين المعلم والمتعلمين أو بين المتعلمين أنفسهم. تعتبر هذه الإستراتيجية من الطرق الفاعلة التي تشجع على التفكير النقدي، وتطوير مهارات التواصل، وتعميق الفهم من خلال النقاشات المفتوحة والموجهة.

طرح سؤال على المتعلمين

ما هي مميزات البرمجة بلغة البايثون؟ (لماذا اخترنا لغة البايثون من بين باقي اللغات البرمجية؟)

- سهولة التعلم والاستخدام.
- تطوير برمجيات معقدة من خلالها.
- تستخدم في مجالات متعددة مثل تطبيقات الويب، الذكاء الاصطناعي.
- يحتوي على العديد من المكتبات الجاهزة.
- تتوافق مع العديد من أنظمة التشغيل.
- التحديث المستمر.

مثال :



الاستراتيجيات التعليمية:

استراتيجية التعلم التعاوني:



التعلم التعاوني هو أسلوب تعليمي يعتمد على تقسيم المتعلمين إلى مجموعات صغيرة للعمل معًا لتحقيق أهداف تعليمية مشتركة. بقصد التعاون فيما بينهم والمشاركة في حل المشكلات، من خلال المشاركة في المعلومات والموارد، مما يعزز التعلم النشط والتفاعل.

تقسيم المتعلمين إلى مجموعات كل مجموعة لا تزيد عن ثلاثة متعلمين ويتم توزيع أسئلة مطبوعة على أوراق يطلب من كل مجموعة حلها مثل كتابة تعليمات:

- إيجاد باقي القسمة.
- إدخال رقمين ثم طباعة ناتج الجمع.
- حساب مساحة الدائرة.

مثال:



الاستراتيجيات التعليمية:

التعلم القائم على حل المشكلات:

التعلم القائم على حل المشكلات هو أسلوب تعليمي يركز على تقديم مشكلات معقدة وواقعية للمتعلمين للعمل على حلها، ويهدف إلى تطوير مهارات التفكير النقدي وحل المشكلات، كما يهدف إلى التعلم الذاتي والعمل الجماعي، حيث يتم تشجيع المتعلمين على البحث والتحليل والتفكير بشكل مستقل وجماعي للتوصل إلى حلول مناسبة.

برنامج تحويل وحدة قياس الزمن من الدقيقة إلى الثانية والعكس.

مشكلة البرنامج:

تحويل وحدة قياس الزمن من الدقيقة إلى الثانية والعكس.

الخوارزمية:

- استقبال من المستخدم قيمة الزمن.
- استقبال من المستخدم وحدة قياس الزمن المطلوب التحويل إليها سواء (دقيقة أو ثانية).
- استخدم المعادلة التالية عند استقبال "M" لتحويل وحدة قياس الزمن بالدقيقة إلى الثانية: $S = (\text{time} * 60)$
- استخدم المعادلة التالية عند استقبال رمز غير "M" لتحويل وحدة قياس الزمن بالثانية إلى الدقيقة:

$$M = (\text{time} / 60)$$

- اعرض قيمة الزمن بعد التحويل لوحدة القياس المطلوبة.

المطلوب:

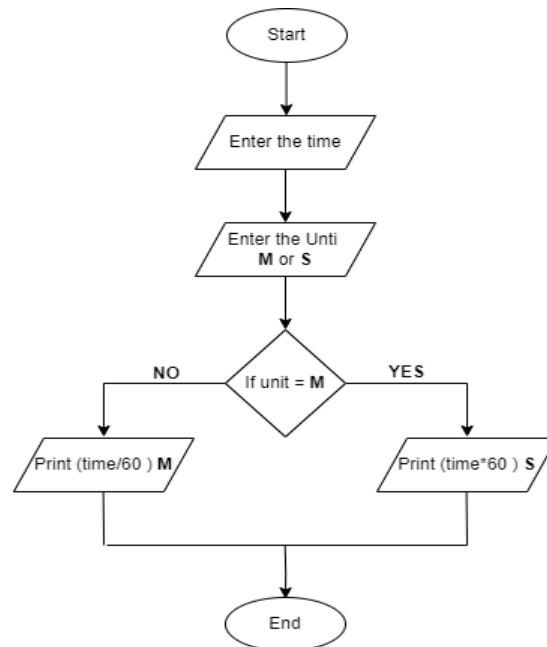
- ارسم خريطة التدفق المناسبة.
- كتابة التعليمات البرمجية اللازمة بشكل صحيح واختبار البرنامج.

مثال:



الاستراتيجيات التعليمية:

خريطة التدفق:



التعليمات البرمجية:

Code

```
1 # Enter the time
2 Time = float(input("Enter the Time : "))
3 # Enter The Time's Unit
4 unit = input ("Enter the Time's Unit M for Minutes or S for Seconds : ")
5 if unit == "M":
6     print (Time * 60, "S")
7 else:
8     print (Time /60, "M")
```

Output

Enter the Time : 90

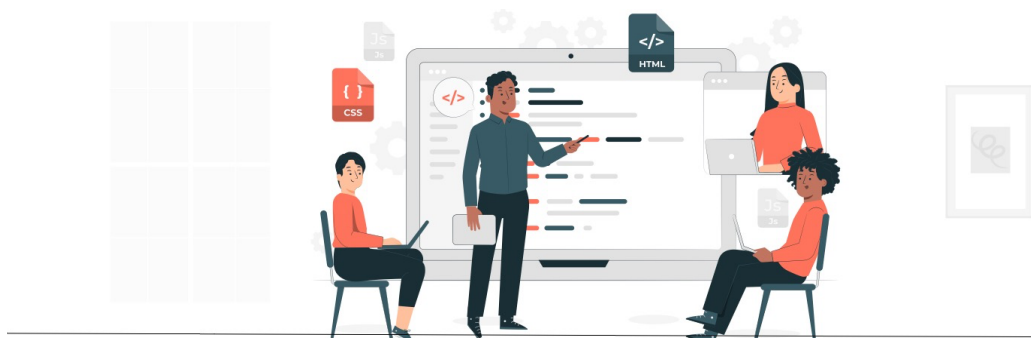
Enter the Time's Unit M for Minutes or S for Seconds : S

1.5 M



الاستراتيجيات التعليمية:

استراتيجية العصف الذهني:



العصف الذهني إستراتيجية تهدف إلى توليد أكبر عدد ممكن من الأفكار حول موضوع معين في بيئة جماعية في إطار وقت محدد، تدعو إلى الإبداع والتفكير الحر بين المتعلمين وتعويدهم على احترام الآراء المختلفة والاستفادة من أفكار الآخرين ومعلوماتهم، ودور المعلم هو التوجيه والإرشاد لتحقيق الأهداف المرجوة.

اكتب التعليمات البرمجية للحصول على الشكل التالي:

مثال:

Code

```
1  # عدد الصفوف
2  n = 4
3
4  for i in range(1, n + 1):
5      # طباعة نجمة واحدة متنوعة بمسافة
6      print("* " * i)
7
```

Output

```
*
* *
* * *
* * * *
```



الاستراتيجيات التعليمية:

استراتيجية التعلم القائم على المشاريع:



التعلم القائم على المشاريع هو نهج تعليمي يركز على إشراك المتعلمين في مشاريع إما في صورة مستقلة أو في إطار تعاوني تتطلب منهم البحث والتخطيط والتنفيذ والتقييم، ويكون دور المعلم هو تقديم التوجيه والإرشاد للمتعلمين من أجل إكمال مشاريعهم بنجاح واكتساب فهم عميق للمفاهيم الأساسية.

1. يقسم المعلم المتعلمين إلى مجموعات ويوضح لهم آلية العمل.
2. تكليف كل مجموعة بتعبئة نموذج التخطيط للمشروع.
3. يبدأ المتعلمين بتنفيذ المهام حسب الخطة الموضوعة.
4. يقيم المعلم مشاريع المجموعات باستخدام استمارة تقييم المشروع.

مثال:

التعلم بالأقران:

التعلم بالأقران إستراتيجية يكتسب فيها المتعلمين مهاراتهم ومعارفهم بمساعدة بعضهم البعض تحت إشراف المعلم، بحيث يكلف المعلم متعلم متفوق في مجال ما بدور الخبير، لتعليم متعلم آخر أقل مستوى منه أو مجموعة معينة ويقودهم نحو فهم الهدف أو إتقان المهارة.

أثناء الحصة الدراسية يتم تقسيم المتعلمين إلى مجموعات ثنائية.
بالرجوع للكتاب، ينفذ المتعلمين معا خطوات إحدى أوراق العمل في الكتاب المدرسي.

مثال:



الاستراتيجيات التعليمية:

استراتيجية الصف المقلوب:



استراتيجية الصف المقلوب هي إستراتيجية تعليمية تعتمد على تزويد المتعلمين بمادة تعليمية حول الدرس كمقاطع فيديو أو ملفات صوتية أو غيرها من الوسائط، للاطلاع عليها في منازلهم أو في أي مكان آخر باستعمال حواسيبهم أو هواتفهم الذكية أو أجهزةهم اللوحية قبل حضور الدرس. في حين يُخصص وقت الدرس للمناقشات والمشاريع والأنشطة.

- يتم تقديم مقاطع فيديو قصيرة تشرح المتغيرات وأنواع البيانات المختلفة في لغة **Python**.
- يُطلب من المتعلمين قراءة مصادر نصية توضح كيفية استخدام المتغيرات وتعديل قيمتها في لغة **Python**.
- في وقت الحصة الدراسية، يتم تنظيم جلسات تطبيقية تهدف إلى تطبيق ما تم مشاهدته في مقاطع الفيديو المرسلة مسبقًا.
- يُطلب من المتعلمين كتابة برامج بسيطة تُستخدم فيه المتغيرات وتُطبق المفاهيم التي تعلموها.
- يقدم المعلم التوجيه أثناء العمل، ويُجيب على الأسئلة.
- يتم توجيه المتعلمين لتصحيح الأخطاء وتحسين أساليبهم في البرمجة.

مثال:

Code

```
1  # تعريف المتغيرات
2  age = 25
3  name = "Ahmed"
4  is_student = True
5  height = 175.5
6
7  # طباعة قيم المتغيرات
8  print("Name:", name)
9  print("Age:", age)
10 print("Is student:", is_student)
11 print("Height:", height)
```

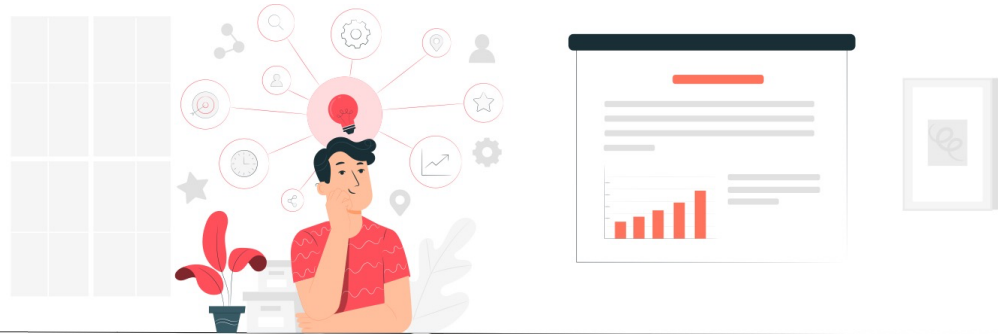
Output

```
Name: Ahmed
Age: 25
Is student: True
Height: 175.5
```



الاستراتيجيات التعليمية:

استراتيجية الخرائط المفاهيمية:



الخرائط المفاهيمية هي إستراتيجية تعليمية فاعلة في تمثيل المعرفة عن طريق أشكال تخطيطية تربط المفاهيم مع بعضها البعض بخطوط أو أسهم يُكتب عليها كلمات تُسمى كلمات الربط لاكتشاف العلاقات بين المفاهيم وتعميق الفهم وتلخيص المعلومات وتقويم الدرس.

مثال: تطبيق استراتيجية الخرائط المفاهيمية في حل مشكلة جمع عددين في لغة Python.

التعليمات البرمجية :

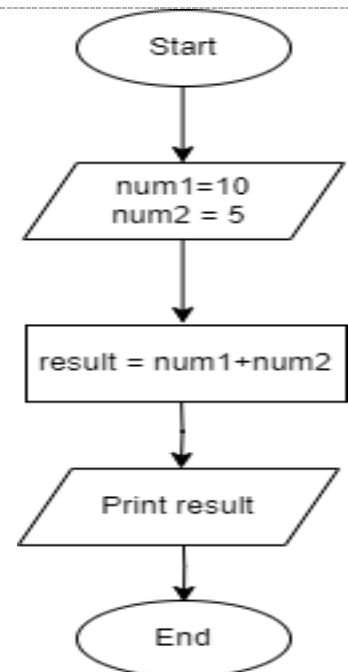
Code

```
1  # تعريف المتغيرات
2  num1 = 10
3  num2 = 5
4
5  # الجمع
6  result = num1 + num2
7
8  # طباعة النتيجة
9  print("الناتج هو:", result)
10
```

Output

الناتج هو: 15

خريطة التدفق :



الوحدة الأولى:

الأمن السيبراني Cyber Security



وصف الوحدة: الأمن السيبراني Cyber Security

في هذه الوحدة، سنتعرف على المفاهيم الأساسية للأمن السيبراني وأمن المعلومات وأهميتها في العصر الرقمي في حماية الأنظمة والمعلومات، وسنتعلم كيفية تطبيق المصادقة الثنائية واستخدام تقنيات الحماية والتشفير لتعزيز أمن البيانات، وسنميز بين الأدوار والمسؤوليات المختلفة ضمن مجالات العمل في وظائف الأمن السيبراني، وسنستكشف التحديات المختلفة التي تواجه الأمن السيبراني وكيفية التعامل معها بفعالية، مما يساعدنا على فهم الدور الحاسم للأمن السيبراني في عالمنا الرقمي المعاصر.

حيث مع التقدم التكنولوجي المتسارع وانتشار الإنترنت في جميع جوانب الحياة اليومية، أصبحت المعلومات والبيانات الرقمية تشكل جزءاً أساسياً من بنيتنا التحتية الحديثة، سواء كان ذلك في القطاعات الحكومية، التجارية، الصحية، أو التعليمية.

واليوم يعتمد العالم بشكل كبير على الأنظمة الرقمية لتخزين ومعالجة وتبادل المعلومات لكن، مع هذا الاعتماد المتزايد تبرز تحديات جديدة تتعلق بحماية هذه المعلومات من الهجمات السيبرانية التي يمكن أن تسبب خسائر مالية هائلة وتهدد الخصوصية والأمن القومي.

وبذلك ممكن تعريف الأمن السيبراني Cybersecurity، بأنه مجال حيوي يهدف إلى حماية الأنظمة والشبكات و البرامج من الهجمات الرقمية، ويشمل مجموعة واسعة من التدابير والتقنيات المصممة للحفاظ على سرية وسلامة وتوافر المعلومات، بالإضافة إلى ضمان عدم تعرضها للاختراق أو السرقة أو التلف باستخدام مجموعة من الأدوات والتقنيات مثل التشفير، والجدران النارية، وأنظمة الكشف عن التسلل ومنع التسلل.

إلى جانب الأدوات والتقنيات، يلعب الوعي الأمني والتدريب دوراً محورياً في تعزيز الأمن السيبراني Cybersecurity. من خلال تدريب الأفراد على التعرف على التهديدات والتعامل معها، وتطبيق سياسات أمنية صارمة مثل تحديثات البرمجيات وإدارة كلمات المرور والنسخ الاحتياطي للبيانات، يمكن تقليل المخاطر وتحسين الحماية بشكل كبير.



خطة الدروس: الوحدة الأولى الأمن السيبراني Cyber Security

 Cyber Security1	حصة دراسية	مفاهيم الأمن السيبراني.	
 Cyber Security2	حصة دراسية	المحاور الرئيسية للأمن المعلومات والأمن السيبراني CIA.	
 Cyber Security3	حصة دراسية	الإجراءات الآمنة للمحاور الرئيسية للأمن السيبراني CIA.	
 Cyber Security4	حصة دراسية	ممارسات الأمن السيبراني.	
		الوظائف في مجال الأمن السيبراني.	
		الجوانب الأخلاقية الرئيسية في مجال الأمن السيبراني.	



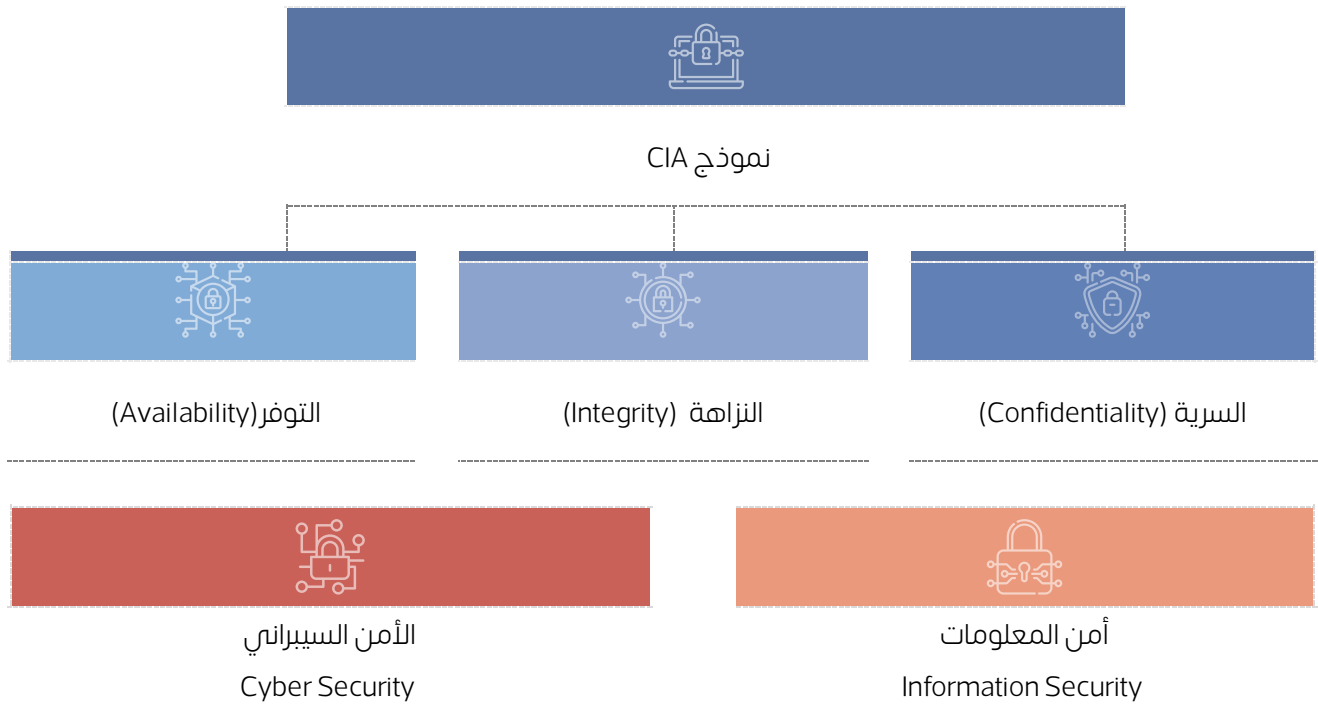


ناتج التعلم: الوحدة الأولى الأمن السيبراني Cyber Security

- 1 شرح مفهوم الأمن السيبراني وأمن المعلومات وأهميتها في العصر الرقمي.
- 2 تفسير عناصر مفهوم CIA (السرية، النزاهة، التوفر)، وتوضيح دورها في حماية الأنظمة والمعلومات.
- 3 تطبيق خطوات المصادقة الثنائية، واستخدام تقنيات الحماية والتشفير لتعزيز أمن البيانات.
- 4 توظيف أدوات مثل برامج إدارة كلمات المرور ووسائل التصفح الآمن لحماية المعلومات الشخصية.
- 5 التمييز بين الأدوار والمسؤوليات المختلفة ضمن مجالات العمل في وظائف الأمن السيبراني.
- 6 إظهار سلوكاً أخلاقياً ومسؤولية عالية عند التعامل مع البيانات الرقمية والمعلومات الحساسة.



مطلحات حاسوبية – Key Words





تابع/ مصطلحات حاسوبية — Key Words



الشبكة الافتراضية الخاصة
Virtual Private Network (VPN)



المصادقة
Authentication



برامج ادارة كلمات المرور
Password Managers



تطبيقات المراسلة الآمنة (التشفير)
Secure Messaging Apps (Encryption)



الخصوصية في متصفحات الإنترنت
Privacy-Focused Web Browsers



تشفير البريد الإلكتروني
Encryption for Email



المحافظ الرقمية
Digital Wallets



طرق الدفع الآمنة
Secure Payment Methods



التصيد الاحتيالي
Phishing



ممارسات الأمن السيبراني
Cybersecurity Practices

حماية الأجهزة
Security

كلمات مرور قوية
Strong Passwords



الإبلاغ
Report



إدارة عملية مشاركة البيانات
Data Sharing



تابع/ مصطلحات حاسوبية — Key Words



مهندس أمن
Security Engineer



ممارسات الأمن السيبراني
Cybersecurity Analyst



المخترق الأخلاقي
Ethical Hacker



مستشار أمني
Security Consultant



المستجيب للحوادث
Incident Responder



مهندس تصميم أنظمة أمنية
Security Architect



إدارة التصحيحات الأمنية
Patch Management



محلل مركز العمليات الأمنية
Security Operations Center Analyst
(SOC Analyst)

النسخ الاحتياطي للبيانات
Data backUp



استراتيجيات تعليمية مقترحة : الأمن السيبراني

استخدام استراتيجية المناقشة والحوار : لاستقطاب انتباه المتعلمين وربط موضوع الأمن السيبراني بموقف حياتي: هل تعرض أحدكم أو أحد معارفكم للاختراق من قبل المخترقين في الواتس آب او الإنستغرام أو أي من برامج التواصل الاجتماعية الأخرى الخاصة بكم؟ حدثنا عنها.

تهدف الاستراتيجية إثارة المواضيع التالي:

- الإجراءات الأمنية مثل المصادقة والشبكات الافتراضية (VPN) وتشفير البيانات وغيرها.
- الممارسات الأمنية السيبراني مثل كلمات السر القوية والمصادقة وحماية الأجهزة وغيرها.
- استخدام التكنولوجيا المسؤول.
- التواصل بأمان.

استخدام استراتيجية التعلم التعاوني : بتقسيم الفصل لمجموعات ثم عرض نماذج للممارسات الخاطئة في استخدام التكنولوجيا وتكليف المجموعات بتصنيفها إذا كانت آمنة أم لا، مثل عرض مجموعة من المواقع أو الإيميلات واختبار المجموعات إذا كان بإمكانها اكتشاف الآمنة منها من الخطيرة.



أوراق العمل : الوحدة الأولى الأمن السيبراني Cyber Security

يتم شرح محتوى أوراق العمل من خلال إعداد فيديو يقدم أثناء الحصة الدراسية على يستفيد منها المتعلم في استخدامه لأجهزته الخاصة من باب التوعية في الأمن السيبراني.

أوراق عمل
(1-2-3) اللاصفية

الوحدة الثانية:

برمجة بايثون Python



وصف الوحدة: برمجة بايثون Python.

في عصرنا الرقمي، تلعب البرمجة دورًا حيويًا في تطوير التكنولوجيا الحديثة التي نعتمد عليها يوميًا. في هذه الوحدة، "برمجة بايثون"، سنتعرف على أهمية مفهوم البرمجة في تحويل الأفكار إلى تطبيقات وأدوات عملية تُستخدم في شتى المجالات من خلال كتابة تعليمات وأوامر لجهاز الحاسوب لتنفيذ مهام محددة مثل تطوير التطبيقات، المواقع الإلكترونية، الألعاب، والأنظمة التكنولوجية الأخرى.

يتبعها مراحل إنشاء برنامج حاسوبي، بدءًا من تحليل المشكلة، مرورًا بكتابة الخوارزمية، وتصميم خريطة تدفق الإجراءات لحل المشكلة، وصولاً إلى كتابة التعليمات البرمجية، ثم سنتعرف على لغة البرمجة بايثون (Python) التي تُعد واحدة من أكثر لغات البرمجة شيوعًا بفضل سهولتها ومرونتها من خلال محرر الأوامر (PyCharm) وهو بيئة تطوير متكاملة تساعد في كتابة وتحرير التعليمات البرمجية بكفاءة، بالإضافة إلى ذلك، سنستكشف أنواع هياكل البيانات في Python، وكيفية استخدامها لتنظيم وتخزين البيانات بطرق فعّالة.

من خلال هذه الوحدة، سيكتسب المتعلم فهمًا عميقًا للمفاهيم الأساسية للبرمجة، مما يمكنهم من تطبيق هذه المعرفة في تطوير حلول برمجية فعّالة، ويفتح أمامهم آفاقًا واسعة في عالم التكنولوجيا.



الدرس الأول : مدخل إلى البرمجة Introduction to programming

خطة الدروس: مدخل إلى البرمجة

 Python1	حصة دراسية	• مدخل الى البرمجة: مفهوم البرمجة. • خطوات انشاء برنامج لغة البرمجة (Python). • كتابة خوارزمية لحل مشكلة. • تصميم خريطة تدفق لإجراء حل مشكله. • استخدام محرر الأوامر (PyCharm).	
 Python2	حصة دراسية	• مدخل الى البرمجة: التعرف على الدالة المدمجة (print). • التعرف على هياكل البيانات. • معرفة أنواع هياكل البيانات في بايثون.	





نواتج التعلم : مدخل إلى البرمجة

- 1 توضيح مفهوم البرمجة وأهميتها في عالم التكنولوجيا.
- 2 تحديد المراحل الأساسية لتطوير البرنامج الحاسوبية.
- 3 كتابة خوارزمية مناسبة لحل مشكلة محددة بطريقة منظمة.
- 4 تصميم خريطة تدفق **Flowchart** توضح خطوات حل المشكلة.
- 5 التعرف على لغة البرمجة **Python** ومجالات استخدامها.
- 6 استخدام بيئة التطوير المتكاملة **PyCharm IDE** لكتابة، وتنفيذ التعليمات البرمجية.
- 7 استخدام الدالة المدمجة **print()** في طباعة المخرجات.
- 8 التمييز بين أنواع البيانات المختلفة في **Python**.



مصطلحات حاسوبية: مدخل إلى البرمجة.



الخوارزمية
Algorithm



البرمجة
Programming



عملية إدخال البيانات
Input



خرائط التدفق
Flowchart



تابع/ مصطلحات حاسوبية: مدخل إلى البرمجة.



عملية المعالجة

Process



عملية إخراج البيانات

Output



لغة البرمجة

Programming Language



اتخاذ قرار

Decision



المفسر

Interpreter



المترجم

Compiler



شجرة المشروع والملفات

Project



محرك الأوامر

PyCharm



محرك التعليمات البرمجية

Code Editor



ملف بايثون

Python file



القوائم الرئيسية

Main Menu



الأمر

Run



هياكل البيانات الأولية

Primitive Data Structures



هياكل البيانات

Data structure



تابع/ مصطلحات حاسوبية: مدخل إلى البرمجة.



الأعداد العشرية

Float



الأعداد الصحيحة

Integer



البيانات المنطقية

Boolean



السلسلة النصية

String



هياكل البيانات الخطية

Linear Data Structures



هياكل البيانات غير الأولية

Non-primitive Data Structures



المجموعة

Set



القائمة

List



الصف

Tuple



القاموس

Dictionary



ملاحظات للمعلم:

شرح واجهة المستخدم: عند شرح نافذة المحرر يجب التطرق إلى مكونات واجهة المستخدم في PyCharm بما في ذلك نافذة المشروع، نافذة الإخراج (console) ، نافذة التوجيه، ونوافذ الأدوات الأخرى.	
حفظ الملف: التأكيد على خطوات حفظ الملف وامتداده <code>.py</code> ، مثلاً <code>pract1.py</code>.	
تخصيص الواجهة: يمكن تخصيص واجهة البرنامج بناءً على احتياجات المتعلمين، مثل تغيير حجم الخط أو تخصيص الألوان لتسهيل القراءة.	
إغلاق المشروع : التأكيد على أهمية إغلاق المشروع لكل متعلم بعد انتهاء جلسة العمل لضمان استمرار كل متعلم في تنفيذ أوراق العمل الخاصة به فقط.	

لتغيير نوع وحجم الخط في PyCharm، يمكنك إتباع الخطوات التالية:



1	اختيار الأمر settings من قائمة file.
2	تحديد الاختيار Editor من القائمة الجانبية ل appearance and behavior من نافذة Settings .
3	اختيار الأمر الفرعي Font من Editor ثم تحديد الخيارات المطلوبة .
4	تطبيق التغييرات بالضغط على Apply .

لتغيير لون خلفية الشاشة في PyCharm، يمكنك إتباع الخطوات التالية:



1	اختيار الأمر settings من قائمة file.
2	تحديد الاختيار Appearance من القائمة الجانبية ل appearance and behavior من نافذة Settings.
3	يتم اختيار ال theme المرغوب.
4	تطبيق التغييرات بالضغط على Apply .



تابع - ملاحظات للمعلم:

عزيزي المعلم، في المراحل التعليمية السابقة تعرف المتعلمين على مبادئ التفكير المنطقي وتصميم خرائط التدفق، والآن حان الوقت لصقل هذه المبادئ وتوظيفها، فالتخطيط الجيد والمنظم يعتبر من الممارسات والعادات السليمة للمبرمجين والتي تساعد المتعلم على تصور عملية سير الأوامر البرمجية بصورة مرئية، وهذا الشيء مهم جداً في ضوء افتقار محرر الأوامر (PyCharm) للواجهة الرسومية. فلابد من التركيز على مهارتين أساسيتين:

1. حل مشكلة ما باستخدام الخوارزميات وترجمتها لخريطة تدفق تدريجياً من البسيط إلى المركب.
2. تدريب المتعلمين على الكتابة على لوحة المفاتيح بصورة جيدة، فكلما زادت طلاقة الكتابة ازداد معها سرعة الحل مما يتيح مساحة أكثر لمزيد من التمارين.

نموذج مقترح لشرح الخوارزميات وخرائط التدفق:

الخطوة الأولى: استقطاب انتباه المتعلمين.

الهدف: استعراض مشكلة وتحليلها وإبداء حلول واقعية لها.

باستخدام استراتيجية المناقشة والحوار أو العصف الذهني، قم بتقديم مشكلة ضعف مجموعة من المتعلمين في العمليات الرياضية وتحديدًا عمليات الضرب.

الهدف: الوصول إلى فكرة تطوير برنامج يساعد المتعلمين على الطلاقة الرياضية في عمليات الضرب البسيطة.

الخطوة الثانية: بناء الخوارزمية.

الهدف: بناء الخطوات المنطقية لحل المشكلة.

باستخدام استراتيجية التعلم التعاوني قم بتقسيم المتعلمين إلى مجموعات للبدء بمناقشة وكتابة الخطوات اللازمة التي يحتاجها المتعلم (المستخدم) لحل مسألة رياضية بسيطة واحدة (ملاحظة: يخصص وقت محدد للمجموعات لأداء المهمة).

عرض ومناقشة المعلم للخوارزميات المقترحة منهم والتأكد من احتوائها على التالي:

1. عرض السؤال للمتعلم.
2. استقبال الإجابة من المتعلم.
3. اختبار الإجابة إذا كانت صحيحة أم خاطئة.
4. عرض النتيجة للمتعلم.



تابع - ملاحظات للمعلم:

الخطوة الثالثة: تصميم خريطة التدفق.

الهدف: تحويل خطوات الحل (الخوارزمية) لخريطة تدفق.

باستخدام استراتيجية المناقشة والحوار واستراتيجية التعليم المباشر حوّل خوارزمية الحل إلى خريطة تدفق، ثم كلف كل فريق بتصميم خريطة التدفق الخاصة بهم (ملاحظة: يخصص وقت محدد للمجموعات لأداء المهمة).

التقويم:

استعرض مشكلة ما واطلب من المجموعات التالي:

1. تحليل المشكلة ووضع الحلول المناسبة.
2. تصميم خوارزمية الحل.
3. تصميم خريطة التدفق.

نموذج مقترح لتدريب المتعلمين على الكتابة باستخدام لوحة المفاتيح:

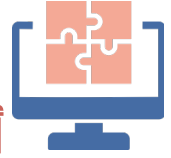
باستخدام استراتيجية التعلم التعاوني قسّم المتعلمين إلى مجموعات واطلب منهم - باستخدام محرر الأوامر - (PyCharm) تصميم برنامج يعرض للمستخدم عند التشغيل الجدول المدرسي الخاص في الفصل كما يلي:

D/P	1	2	3	4	5	6	7
Sunday	subject	subject	subject	subject	Subject	subject	Subject
Monday	subject	subject	subject	subject	Subject	subject	Subject
Tuesday	subject	subject	subject	subject	Subject	subject	Subject
Wednesday	subject	subject	subject	subject	Subject	subject	Subject
Thursday	subject	subject	subject	subject	Subject	subject	Subject

ولخلق روح المنافسة، حوّل التمرين إلى مسابقة بين المجموعات وخصص لوحة ثابتة في الفصل تستعرض كل الفصول وأسماء المتعلمين الذين ينتمون للمجموعة الفائزة في المسابقة.

قوانين المسابقة:

1. يقوم كل عضو في الفريق بتصميم مشروعه الخاص.
2. يمنع الكتابة عن أي عضو وإلا تُستبعد المجموعة.
3. تعتبر المجموعة قد نفذت المهمة عندما ينتهي جميع الأعضاء من تصميم مشاريعهم.
4. ترتيب المجموعات الفائزة يكون حسب سرعة إنجاز المهمة.



أوراق العمل: مدخل إلى البرمجة

ورقة عمل (4): برنامج حساب مساحة الدائرة.

المشكلة:

من خلال دراستك لقوانين مساحة الأشكال الهندسية، احسب مساحة الدائرة بمعلومية نصف القطر.

المدخلات: نصف القطر r .

المعالجة: حساب مساحة الدائرة باستخدام المعادلة.

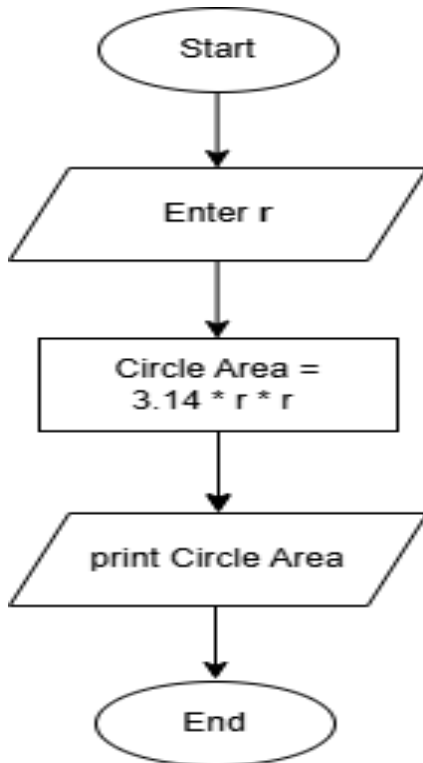
مساحة الدائرة $= \pi r^2$ حيث $\pi = 3.14$

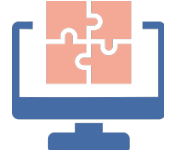
المخرجات: طباعة مساحة الدائرة.

الخوارزمية:

1. بداية البرنامج.
2. استقبال من المستخدم قيمة نصف قطر الدائرة (r).
3. حساب مساحة الدائرة باستخدام المعادلة:
مساحة الدائرة πr^2 حيث $\pi = 3.14$
4. عرض مساحة الدائرة على الشاشة.
5. إنهاء البرنامج.

خريطة التدفق Flowchart :

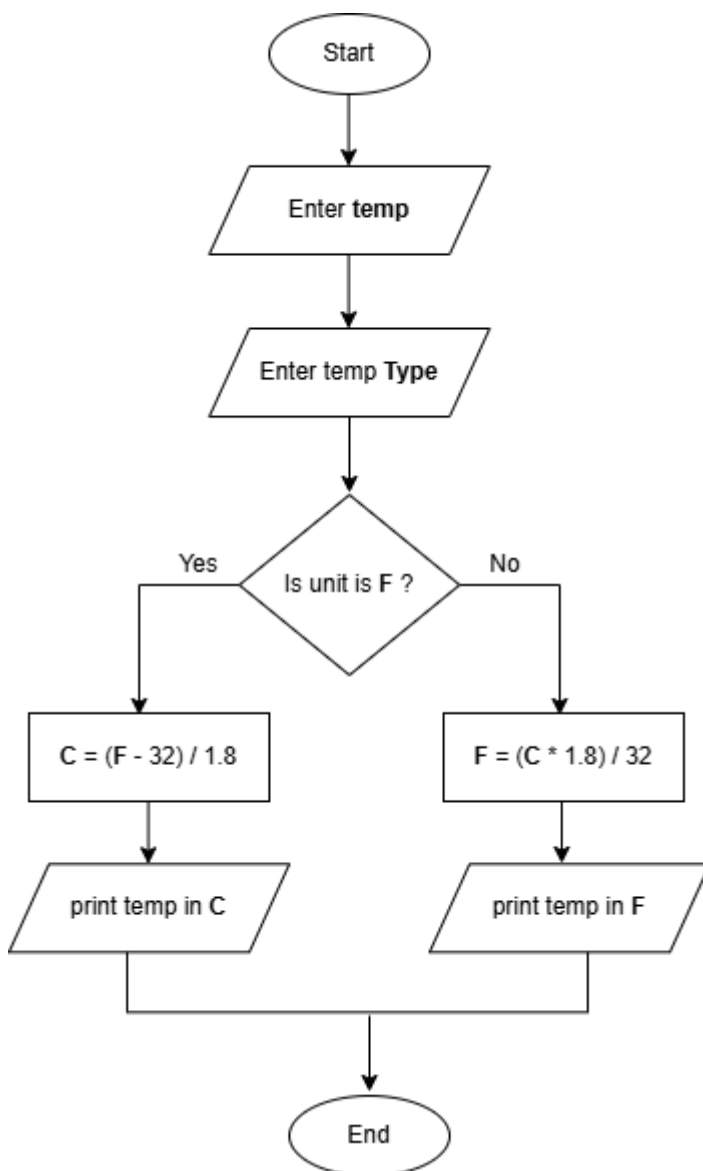




تابع / أوراق العمل: مدخل إلى البرمجة

ورقة عمل (5): برنامج تحويل وحدات قياس درجات الحرارة

خريطة التدفق Flowchart



المشكلة:

التحويل ما بين وحدات القياس المختلفة لدرجة الحرارة.

المدخلات: درجة الحرارة temp ووحدة قياس درجة الحرارة type.

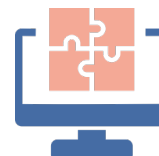
المعالجة: اتخاذ القرار ثم:

- حساب درجة الحرارة باستخدام المعادلة التالية للتحويل درجة الحرارة من فهرنهايت إلى سيليزية $C = (F - 32) \div 1.8$
- استخدام المعادلة التالية للتحويل درجة الحرارة من سيليزية إلى فهرنهايت: $F = (C \times 1.8) + 32$

المخرجات: عرض درجة الحرارة بعد التحويل لوحدة القياس المطلوبة.

الخوارزمية:

1. بداية البرنامج.
2. استقبال من المستخدم درجة الحرارة الطقس.
3. استقبال من المستخدم نوع وحدة القياس (سيليزي أو فهرنهايت).
4. استخدام المعادلة التالية للتحويل درجة الحرارة من فهرنهايت إلى سيليزية: $C = (F - 32) \div 1.8$
5. استخدام المعادلة التالية لغير ذلك: $F = (C \times 1.8) + 32$
6. عرض درجة الحرارة بعد التحويل لوحدة القياس المطلوبة.
7. نهاية البرنامج.



تابع / أوراق العمل: مدخل إلى البرمجة

مقترح أمثلة أخرى:

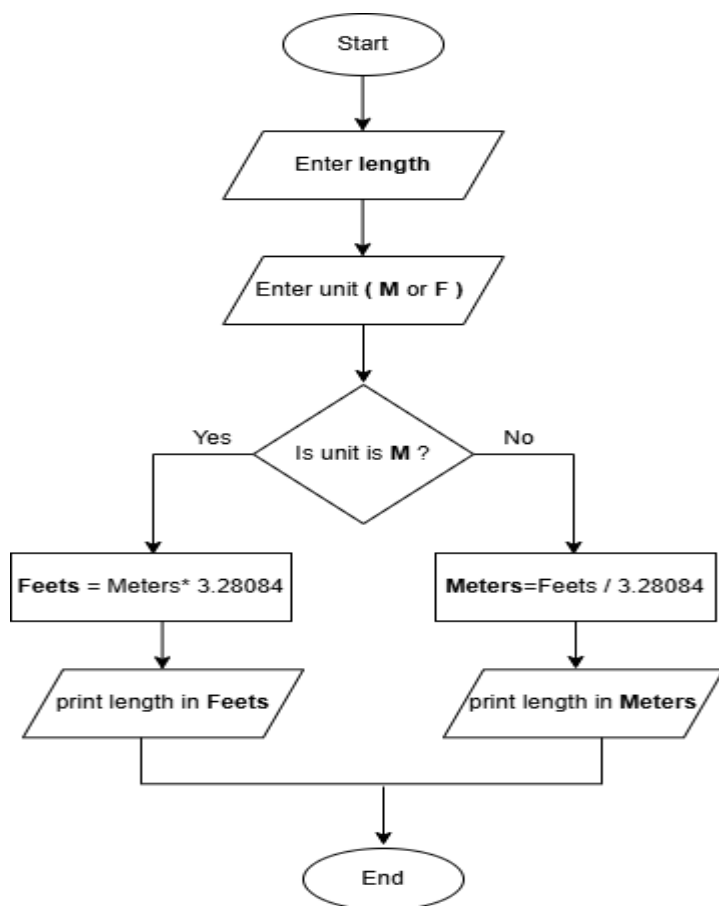
المشكلة:

التحويل ما بين وحدات القياس المختلفة للأطوال

الخوارزمية

1. تحويل الأطوال بين الأمتار والقدم.
2. استقبل من المستخدم الطول.
3. استقبل من المستخدم نوع الطول المدخل.
4. استخدم المعادلة إذا كان الطول بالأمتار:
$$\text{Feet} = \text{Meters} \times 3.28084$$
5. استخدم المعادلة إذا كان غير ذلك:
$$\text{Meters} = \text{Feet} \div 3.28084$$
6. اعرض الأطوال بعد التحويل لوحدة القياس المطلوبة.

خريطة التدفق : Flowchart



التعليمات البرمجية :

Code

```
1 # استقبل القيم من المستخدم
2 length = float(input("Enter the length: "))
3 unit = input("Enter the unit F or M: ")
4
5 if unit == "M":
6     feet = length * 3.28084
7     print(feet)
8 else:
9     meters = length / 3.28084
10    print(meters)
```

Output

```
"/Users/mirror/PycharmProjects/Grade 10/.ve
Enter the length: 10
Enter the unit F or M: F
3.047999902464003

Process finished with exit code 0
```



الدرس الثاني: المتغيرات Variables

البرمجة باستخدام لغة بايثون تتيح لنا القدرة على إنشاء برامج متنوعة باستخدام تعليمات واضحة وبسيطة، في بداية تعلم البرمجة من المهم التعرف على بعض المفاهيم الأساسية مثل المتغيرات وأنواعها، وكيفية تخصيص القيم لها، حيث أن المتغيرات هي أماكن في الذاكرة يمكن تخزين البيانات فيها، ويمكن أن تكون هذه البيانات من أنواع مختلفة مثل الأعداد الصحيحة `integer`، الأعداد العشرية `float`، النصوص `string`، والقيم المنطقية `Boolean` ومن المهم الحرص على تسمية المتغيرات بطريقة صحيحة وفقاً لقواعد التسمية في لغة `Python`.

سنتعلم في هذه الوحدة كيفية استخدام الدالة `type()` للتعرف على نوع المتغير، والدالة `print()` لعرض المعلومات على الشاشة، ودالة `input()` لاستقبال المدخلات من المستخدم، سنتعرف أيضاً على رموز العمليات الحسابية في لغة البايثون وأولويات تنفيذها، وأخيراً سنتعلم على كيفية تحويل البيانات باستخدام الدوال المناسبة مثل `int()`، `float()` و `str()` حيث أن هذه المفاهيم الأساسية تشكل الأساس لكتابة برامج فعالة ومفيدة في لغة `Python`.

خطة الدروس: المتغيرات Variables



Variables1



حصة دراسية

- مفهوم المتغيرات و الإعلان عن متغير.
- تحديد أنواع المتغيرات.
- قواعد تسمية المتغيرات.
- استخدام المتغيرات.
- استخدام دالة `type()`.
- استخدام دالة `print()`.
- استخدام دالة `input()`.
- التحويل بين أنواع البيانات.





ناتج التعلم: المتغيرات Variables

1 توضيح مفهوم المتغيرات ودورها في تخزين البيانات داخل البرنامج.

2 الإعلان عن متغيرات باستخدام الصيغة الصحيحة وفق قواعد لغة Python.

3 شرح قواعد تسمية المتغيرات وتطبيقها بشكل سليم عند كتابة التعليمات البرمجية.

4 استخدام المتغيرات بفاعلية في تنفيذ العمليات البرمجية.

5 توظيف الدالة المدمجة `print()` في طباعة القيم والمخرجات على الشاشة.

6 استخدام الدالة `input()` لاستقبال بيانات من المستخدم.

7 تحويل البيانات بين الأنواع المختلفة بما يتناسب مع متطلبات البرنامج.



مصطلحات حاسوبية: المتغيرات Variables



إنشاء المتغيرات
Create Variables



متغير
Variable



بيانات نصية
string



دالة
Function



بيانات عددية عشرية
Float



بيانات عددية صحيحة
Integer



تابع/ مصطلحات حاسوبية: المتغيرات Variables



دالة الإخراج
الطباعة print



بيانات منطقية
Boolean



دالة نوع المتغير
Type



دالة الادخال
Input



استراتيجيات تعليمية مقترحة : المتغيرات Variables

استخدام استراتيجية المناقشة والحوار : يدير المعلم نقاشاً مع المتعلمين حول أمثلة من الحياة اليومية، مثل تخزين نتيجة للعب في لعبة إلكترونية (Score) أو عدد النقاط التي يجمعها الفريق، ثم يوجههم لاستنتاج أن المتغيرات تُستخدم لحفظ القيم التي قد تتغير أثناء تشغيل البرنامج.

استخدام استراتيجية التعلم التعاوني : للتعرف على أنواع البيانات بتقسيم المتعلمين لمجموعات ثم عرض صورة وهمية للبطاقة المدنية وتكليفهم باستخراج البيانات وتدوين أنواعها.





أوراق العمل: المتغيرات Variables

ورقة عمل (6): برنامج رحلة إلى أبراج الكويت

المشكلة: حساب تكلفة رحلة مدرسية لأبراج الكويت بمعلومية تكلفة المواصلات، دخول الأبراج والوجبة.

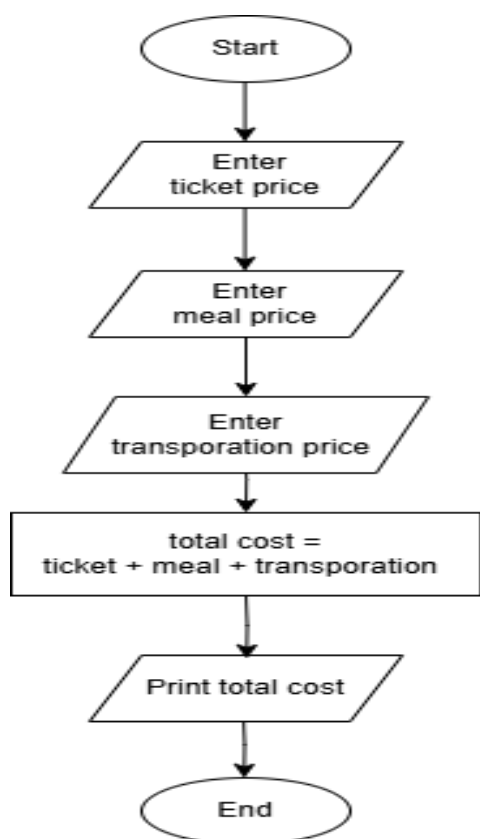
المدخلات: سعر دخول أبراج الكويت *ticket* وسعر الوجبة *meal* وسعر المواصلات *transportation*.

المعالجة: حساب إجمالي تكلفة الرحلة *total_cost* باستخدام المعادلة:

$$\text{total_cost} = \text{ticket} + \text{meal} + \text{transportation}$$

المخرجات: عرض إجمالي تكلفة الرحلة *total_cost*.

خريطة التدفق FlowChart :



الخوارزمية :

1. بداية البرنامج.
2. استقبال من المستخدم:
 - أ. سعر دخول أبراج الكويت.
 - ب. سعر الوجبة.
 - ج. سعر المواصلات.
3. حساب إجمالي تكلفة الرحلة.
4. طباعة إجمالي تكلفة الرحلة.
5. نهاية البرنامج.



تابع/ أوراق العمل: المتغيرات Variables

تابع/ ورقة عمل (6): برنامج رحلة إلى أبراج الكويت

التعليمات البرمجية:

Code

```
1 # Get ticket price.
2 ticket = float (input ('Enter ticket price:'))
3 # Get meal price.
4 meal = float (input ('Enter meal price:'))
5 # Get transportation price.
6 transportation = float (input ('Enter transportation price:'))
7 total_coast = ticket + meal + transportation
8 print('The total coast for Kuwait towers tripis',total_coast,'KD')
```

Output

```
Enter ticket price:3
Enter meal price:1.500
Enter transportation price:2
The total coast for Kuwait towers tripis 6.5 KD
```



تابع - أوراق العمل: المتغيرات Variables.

ورقة عمل (7): حساب المبلغ المستحق لحقائب السفر

المشكلة: حساب المبلغ المستحق لحقائب السفر التي تخطت الوزن المسموح به لرحلة الطيران.

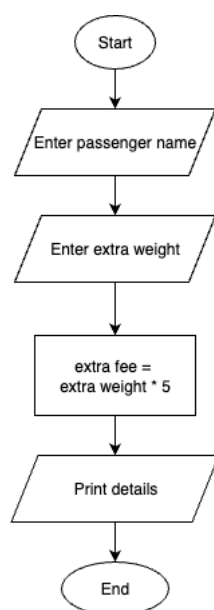
المدخلات: اسم المستخدم `passenger_name`, الوزن الإضافي `extra_weight`.

المعالجة: حساب المبلغ المستحق باستخدام المعادلة:

المبلغ المستحق `extra_fee` = الوزن الإضافي `extra_weight` x 5 KD

المخرجات: طباعة التفاصيل.

خريطة التدفق FlowChart:



الخوارزمية:

1. بداية البرنامج.
2. استقبال من المستخدم اسم المسافر `passenger_name`.
3. استقبال من المستخدم قيمة الوزن الإضافي `extra_weight` بالكيلوجرام.
4. حساب المبلغ المستحق `extra_fee` بوحدة الدينار الكويتي باستخدام المعادلة:
$$extra_fee = extra_weight \times 5$$
5. طباعة التفاصيل على الشاشة.
6. نهاية البرنامج.



تابع / أوراق العمل: المتغيرات Variables

تابع / ورقة عمل (7): حساب المبلغ المستحق لحقائب السفر

التعليمات البرمجية:

Code

```
1 #Get passenger name.
2 passenger_name = input('Enter passenger name: ')
3 #Get extra bag weight in kg (float).
4 extra_weight = float( input('Enter extra weight: '))
5 #Calculate extra fee.
6 extra_fee = extra_weight * 5
7 # Print Details
8 print('Dear', passenger_name, ', your extra fee is', extra_fee , 'KD')
```

Output

```
Enter passenger name: Bader
Enter extra weight: 6
Dear Bader , your extra fee is 30.0 KD
```

ملاحظة:

الدالة المدمجة (input) ترجع دائماً قيمة نصية، فيتوجب تحويل هذه القيمة النصية إلى قيمة عددية لإجراء العمليات الحسابية.



الدرس الثالث: السلاسل النصية Strings.

السلاسل النصية في البرمجة تمثل تسلسلاً من الأحرف أو الرموز المتتالية داخل علامة اقتباس (سواء كانت فردية أو مزدوجة) في لغة البرمجة، هذه السلاسل تُستخدم لتمثيل النصوص والبيانات التي تكون من الأحرف مثل الأرقام والرموز الخاصة، تتميز بعدة خصائص تجعل التعامل معها مرناً وسهلاً، يمكن الوصول إلى أحرف محددة داخل السلسلة النصية باستخدام الفهرسة (Indexing) والقطع (Slicing) لاستخراج أجزاء محددة من النص أيضاً عمليات جمع السلاسل النصية (Concatenation) تسمح بدمج نصين أو أكثر و تكرار السلاسل النصية لأكثر من مرة ، أيضاً مقارنة السلاسل النصية لمعرفة مدى تطابقها أو ترتيبها، توفر دوال مفيدة مثل `len()` لمعرفة طول السلسلة النصية، `capitalize()` لجعل الحرف الأول كبيراً، `lower()` - `upper()` لتحويل النص إلى أحرف كبيرة أو صغيرة، `find()` للبحث عن نص معين داخل النص، `replace()` لاستبدال جزء من النص و لإمكانية تنسيق النصوص تستخدم `f-string`، بينما تسمح دالتي `float()` - `int()` بتحويل النصوص إلى أعداد صحيحة وعشرية و دالة `count()` تُستخدم لعد تكرار جزء معين من النص، و `center()` لمحاذاة النص في المنتصف ، وغيرها من الدوال المفيدة .



خطة الدروس: السلاسل النصية Strings.

 Strings1	حصة دراسية	- تعريف السلاسل النصية وخصائصها - عمليات السلاسل النصية - الفهرسة indexing. - القطع Slicing. - جمع السلاسل النصية Concatenation. - تكرار السلاسل النصية. - المقارنة Comparison.	
 Strings2	حصة دراسية	دوال السلاسل : - طول السلسلة النصية len(). - capitalize () - upper() - lower() - find() - replace() - f-string - int() - float() - count() - center()	





نواتج التعلم: السلاسل النصية Strings.

- 1 التمييز بين علامات الاقتباس المستخدمة في إنشاء السلاسل النصية.
- 2 استخدام عملية التقطيع للوصول إلى أجزاء محددة من السلسلة النصية.
- 3 تطبيق دوال شائعة لمعالجة السلاسل النصية.
- 4 تنفيذ عمليات المقارنة بين السلاسل النصية باستخدام المعاملات المنطقية.
- 5 معالجة مدخلات المستخدم النصية، وتحويلها حسب الحاجة إلى تنفيذ تعليمات برمجية.
- 6 إنشاء برامج بسيطة تتضمن مدخلات ومخرجات تعتمد على معالجة السلاسل النصية.



مصطلحات حاسوبية: السلاسل النصية Strings



الفهرسة Indexing



السلاسل النصية string



جمع السلاسل النصية Concatenation



القطع Slicing



الأساليب المدمجة Built-in Methods



الدوال المدمجة Built-in Functions



دالة Len



المقارنة Comparison



تابع/مصطلحات حاسوبية: السلاسل النصية Strings



دالة Lower



دالة Upper



دالة Find



دالة Capitalize



دالة f-string



دالة Replace



دالة Count



دالة Center



استراتيجيات تعليمية مقترحة: السلاسل النصية Strings.

استخدام استراتيجيات التعلم التعاوني: للتعرف على مفهوم الفهرسة والاستقطاع بحيث يقوم المعلم بتقسيم المتعلمين إلى مجموعات لهم فقرة محددة مثل: أبيات شعر، أو سورة كريمة على أن تكون الحروف مرقمة (مفهرسة) ثم يقوم بتكليف كل مجموعة باستخراج كلمات محددة بتحديد رقم الحرف الأول والأخير، مثال:

					14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						د	ح	أ		ا	ل	ل	ا		و	هـ		ل	ق
									25	24	23	22	21	20	19	18	17	16	15
										د	م	ص	ل	ا		ا	ل	ل	ا
			42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26
				د	ل	و	ي		م	ل		و		د	ل	ي		م	ل
62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43
ب	ح	أ		أ	و	ف	ك		ا	ل		ن	ك	ي		م	ل		و

استخرج من السورة الكريمة ما يلي:

- كم عدد مرات تكرار لفظ الجلالة "الله" ؟ (شرح مفهوم count).
- ابحث عن أول كلمة "أحد" موضحاً موضع الكلمة بدلالة رقم أول حرف. (شرح مفهوم find).
- كم عدد حروف السورة الكريمة ؟ (شرح مفهوم len).



أوراق العمل: السلاسل النصية Strings.

ورقة عمل (8): تأمين كلمة المرور

المشكلة:

عكس ترتيب الأحرف في السلسلة النصية لتصبح مؤمنة (كلمة السر).

المدخلات:

كلمة المرور (password)

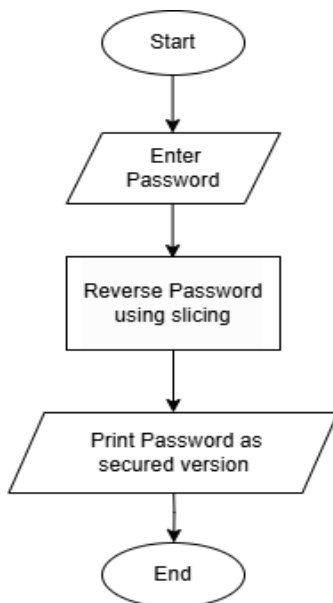
المعالجة:

استخدام التقطيع Slicing لعكس الكلمة وتخزينها في (reversed_password).

المخرجات:

طباعة الكلمة المعكوسة (reversed_password).

خريطة التدفق FlowChart:



الخوارزمية:

1. بداية البرنامج.
2. طلب من المستخدم إدخال كلمة المرور (password).
3. تخزين الكلمة في متغير (password).
4. إنشاء متغيراً جديداً (reversed_password) لتخزين الكلمة المعكوسة.
5. استخدام التقطيع Slicing لعكس الكلمة [::-1].
6. طباعة الكلمة المعكوسة على أنها كلمة مرور محمية.
7. نهاية البرنامج.



تابع/ أوراق العمل: السلاسل النصية Strings.

تابع/ ورقة عمل (8): تأمين كلمة المرور

التعليمات البرمجية:

هذه الطريقة تعتمد على [start: end: step] للوصول إلى أجزاء معينة من السلسلة النصية. مع تعيين قيمة step بقيمة 1- لعكس النص بأكمله.

Code

```
1 # Get the password.
2 password = input("Enter password: ")
3 # Reversed the password.
4 reversed_password = password[::-1]
5 # Print the reversed password.
6 print("Protected password:", reversed_password)
```

Output

```
Enter password: Kuwait
Protected password: tiawuK
```

يمكن إضافة حل آخر لنفس ورقة العمل بعد درس ال loops.



تابع/ أوراق العمل: السلاسل النصية Strings.

ورقة عمل (9) : تقطيع السلسلة النصية

المشكلة:

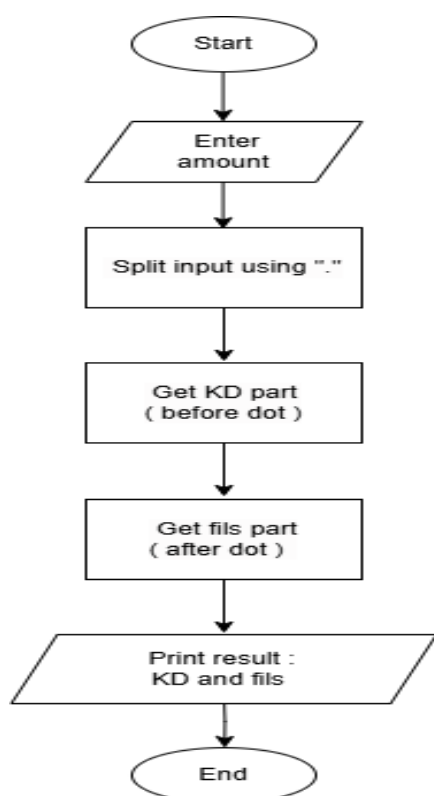
استخراج الجزء الصحيح من السلسلة (دينار)، والجزء العشري (فلس) من مبلغ مالي عددي مُدخل من المستخدم.

المدخلات: المبلغ المالي **amount**.

المعالجة: تحديد موضع الفاصلة العشرية للسلسلة النصية في المتغير **position**، وتحديد جزء قبل الفاصلة المبلغ بالدينار (**Kd**) ، تحديد جزء بعد الفاصلة المبلغ بالفلس (**fls**).

المخرجات: طباعة المبلغ بالتنسيق المطلوب (**KD** و **fls**).

خريطة التدفق FlowChart :



الخوارزمية :

1. بداية البرنامج.
2. استقبال قيمة المبلغ المالي مثال (500.750) من المستخدم.
3. تحديد موضع الفاصلة العشرية للسلسلة النصية المُدخلة.
4. تحديد جزء قبل الفاصلة المبلغ بالدينار.
5. تحديد جزء بعد الفاصلة المبلغ بالفلس.
6. طباعة المبلغ بكتابة بالتنسيق التالي: 500 KD 750 fls and
7. نهاية البرنامج.



تابع/ أوراق العمل: السلاسل النصية Strings.

تابع/ ورقة عمل (9) : تقطيع السلسلة النصية

التعليمات البرمجية:

Code

```
1 var_amount = input("Enter amount e.g (500.750) : ")
2 var_position= var_amount.find('.')
3 var_kd = var_amount [:var_position]
4 var_fils = var_amount [var_position+1:]
5 print(f"{var_kd} KD And {var_fils} fils")
```

Output

```
Enter amount e.g (500.750) : 100.250
100 KD And 250 fils
```



تابع/ أوراق العمل: السلاسل النصية Strings.

ورقة عمل (10): تقطيع السلسلة النصية

المشكلة:

تقسيم الاسم الكامل **full name** (باللغة الانجليزية) للحصول على الاسم الأول والأوسط والأخير.

المدخلات: الاسم الكامل (ثلاثي باللغة الانجليزية) **full_name**.

المعالجة:

- استخدام دالة **len()** لحساب طول السلسلة النصية للاسم الكامل.
- استخدام دالة **find()** ل:
 - إيجاد موضع المسافة الأولى بين الاسم الأول والثاني وحفظه في المتغير **position_space1**
 - إيجاد موضع المسافة الثانية بين الاسم الثاني والثالث وحفظه في المتغير **position_space2**
- استخدام الفهرسة والتقطيع للحصول على الاسم الأول والثاني والثالث وحفظهم في المتغيرات **first_name** و **second_name** و **third_name**.
- استخدام الدالتين **print()** و **capitalize()** لطباعة الاسم الأول والاسم الثاني والاسم الثالث، مع تحويل أول حرف فقط من كل جزء من أجزاء الاسم الكامل إلى حرف كبير (**Capital Letter**).

المخرجات:

- طباعة عدد أحرف الاسم الكامل باستخدام دالة **len()**.
- طباعة الاسم الأول **first_name** مع استخدام الدالة **capitalize()** لتحويل أول حرف من الكلمة إلى **Capital letter**.
- طباعة الاسم الثاني **second_name** مع استخدام الدالة **capitalize()** لتحويل أول حرف من الكلمة إلى **Capital letter**.
- طباعة الاسم الثالث **third_name** مع استخدام الدالة **capitalize()** لتحويل أول حرف من الكلمة إلى **Capital letter**.



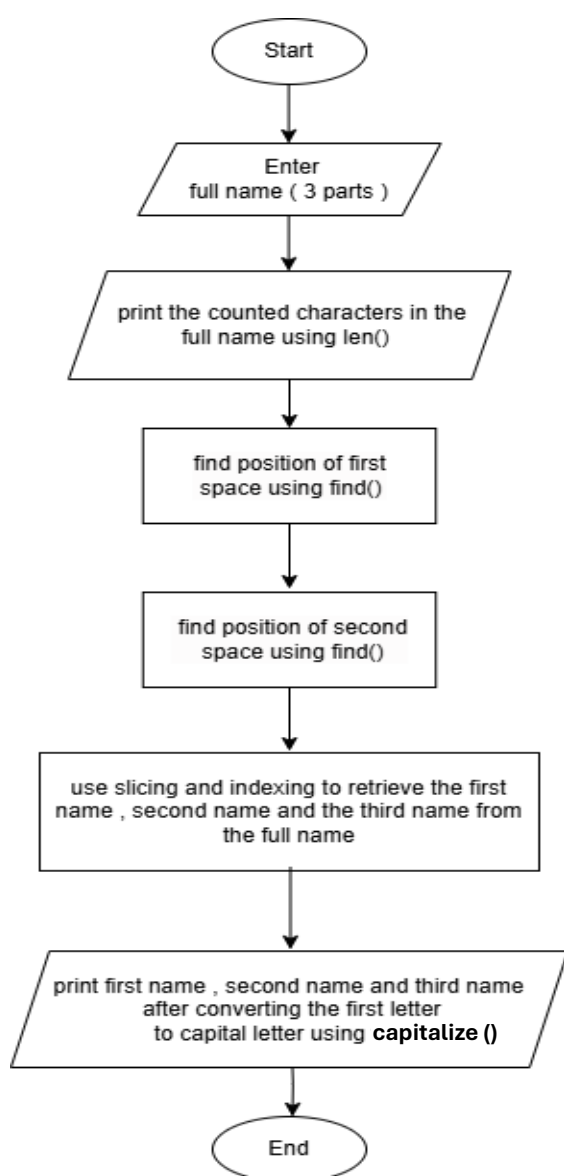
تابع/ أوراق العمل: السلاسل النصية Strings.

تابع/ ورقة عمل (10): تقطيع السلسلة النصية

الخوارزمية :

1. بداية البرنامج.
2. استقبال الاسم الكامل (ثلاثي) باللغة الإنجليزية من المستخدم، وحفظه في متغير `full_name`.
3. طباعة عدد أحرف الاسم الكامل باستخدام دالة `len()`.
4. استخدام دالة `find()` لإيجاد موضع (Position) المسافة الأولى بين الاسم الأول والثاني، وحفظه في متغير `position_space1`.
5. استخدام دالة `find()` لإيجاد موضع (Position) المسافة الثانية بين الاسم الثاني والثالث، وحفظه في متغير `position_space2`.
6. استخدام الفهرسة والتقطيع للحصول على الاسم الأول ، الثاني، والثالث وحفظهم في متغيرات `first_name` و `second_name` و `third_name`.
7. طباعة:
الاسم الأول والاسم الثاني والاسم الثالث مع تحويل أول حرف فقط من كل جزء في الاسم الكامل إلى حرف كبير `Capital letter`.
8. نهاية البرنامج.

خريطة التدفق FlowChart :





تابع/حل أوراق العمل: السلاسل النصية Strings.

تابع/ ورقة عمل (10): تقطيع السلسلة النصية

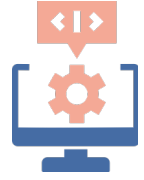
التعليمات البرمجية :

Code

```
1 full_name = input('Enter your full name (Three Parts): ')
2 print('Characters in your name:', len(full_name))
3 position_space1 = full_name.find(' ')
4 position_space2 = full_name.find(' ', position_space1 + 1)
5 first_name = full_name[:position_space1]
6 second_name = full_name[position_space1 + 1:position_space2]
7 third_name = full_name[position_space2+1:]
8 print(first_name.capitalize())
9 print(second_name.capitalize())
10 print (third_name.capitalize())
```

Output

```
Enter your full name (Three Parts): ali mohammad abdullah
Characters in your name: 21
Ali
Mohammad
Abdullah
```



الدرس الرابع : الشروط Conditions

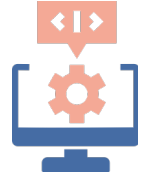
تعتبر عمليات المقارنة مثل '==', '<', '>', '!=', '!' في لغة بايثون أدوات أساسية لاتخاذ القرارات. حيث أن عمليات المقارنة تسمح لنا بمقارنة القيم، بينما تتيح لنا عمليات 'and' - 'or' دمج الشروط المنطقية. وباستخدام الدوال الشرطية 'if', 'elif', 'else' يمكننا تنفيذ أكواد مختلفة بناءً على الشروط المحددة، مما يجعل البرامج أكثر تفاعلية وديناميكية. على سبيل المثال: يمكننا تحديد قيمة معينة واستخدام الشروط لطباعة رسائل مختلفة بناءً على هذه القيمة، مما يساعد في التحكم في تدفق البرنامج بمرونة وفعالية.



خطة الدروس: الشروط Conditions

	حصة دراسية	الشروط : يميز بين عمليات المقارنة والعمليات المنطقية.	
	حصة دراسية	الشروط : يوظف العمليات الشرطية if - else - elif لاتخاذ القرار.	





نواتج التعلم: الشروط Conditions

- 1 توضيح مفهوم الشروط ودورها في اتخاذ القرارات داخل البرنامج .
- 2 التمييز بين عمليات المقارنة (مثل: `<` , `>` , `==`), والعمليات المنطقية (`and` , `or` , `not`) وتحديد استخداماتها.
- 3 توظيف التعليمات الشرطية `if`, `elif`, `else` في كتابة برامج تتخذ قرارات مختلفة حسب المدخلات.
- 4 تصميم برامج تعتمد على شروط متعددة لتنفيذ أوامر مختلفة.
- 5 استخدام الشروط المنطقية المتداخلة في مواقف تتطلب أكثر من شرط.
- 6 التحقق من صحة الشروط، وتنفيذ التصحيح اللازم عند ظهور نتائج غير متوقعة.
- 7 تقدير أهمية التفكير المنطقي في تصميم البرامج واتخاذ القرارات البرمجية الصحيحة.



مصطلحات حاسوبية: الشروط Conditions



عمليات المقارنة

Comparison Operators



التعليمات الشرطية

Condition Statements



عامل المقارنة `==`

Comparison



المسافة البادئة

Indentation



الشرط

Condition



العمليات المنطقية

Logical Operators



عامل التخصيص `=`

Assignment



الكتلة البرمجية

Block of Code



استراتيجيات تعليمية مقترحة: الشروط Conditions

استخدام استراتيجية العصف الذهني : للتعرف على مفهوم الشروط وكيفية استخدامها، عزيزي المعلم ناقش مع المتعلمين فكرة تصميم برنامج يحسب الوقت الذي يستغرقه المتعلم للوصول إلى المدرسة إذا علمت أن المتعلم يستغرق 15 دقيقة من مغادرة المنزل إلى الوصول للمدرسة من دون أي عوائق قد تؤدي لتأخيره.

- قم بتوليد مجموعة من العوائق التي قد تؤدي لتأخير المتعلم مع إعطاء قيمة بالدقيقة لمقدار التأخير.
- مستعيناً بما سبق، صمم برنامج يطرح على المستخدم مجموعة من العوائق التي قد يتعرض لها المتعلم على أن تكون الإجابة بنعم أو لا، ثم إظهار الوقت الكلي الذي يستغرقه المتعلم للوصول إلى المدرسة بناءً على تلك الإجابات.



أوراق العمل: الشروط Conditions

ورقة عمل (11): برنامج حساب الخصومات على المنتجات

المشكلة: تصميم برنامج يحسب التكلفة النهائية لشراء منتجات، وفقاً لقائمة الخصومات المعلنة.

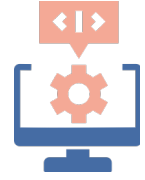
المدخلات: السعر الإجمالي للمنتجات **price**.

المعالجة:

- التحقق من قيمة السعر الإجمالي للمنتجات وتخزين قيمة الخصم بناءً على سعر المنتج في المتغير **discount**.
- حساب السعر النهائي بعد تطبيق الخصم في المتغير **final_cost** كما في المعادلة:

$$\text{final_cost} = \text{price} - (\text{discount})$$

المخرجات: طباعة السعر النهائي **final_cost**.



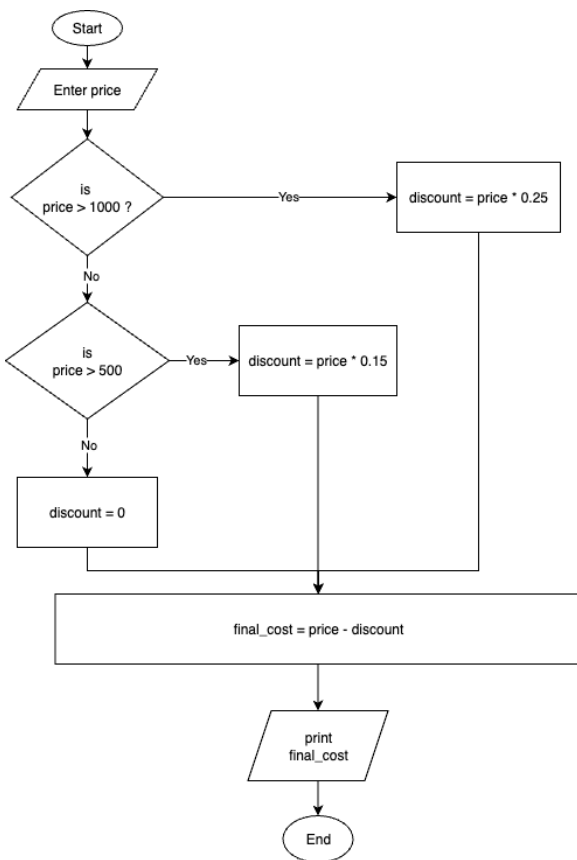
تابع/ أوراق العمل: الشروط Conditions

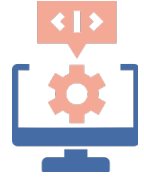
تابع/ ورقة عمل (11): برنامج حساب الخصومات على المنتجات

الخوارزمية :

1. ابداء البرنامج.
2. استقبال من المستخدم السعر الإجمالي للمنتجات.
3. التحقق من قيمة السعر الإجمالي:
 - إذا كان سعر المنتج أكبر من 1000:
 - احسب الخصم بنسبة 25% باستخدام المعادلة:
$$\text{discount} = \text{price} * 0.25$$
 - إذا كان السعر الإجمالي يتراوح ما بين 500 و 1000:
 - احسب الخصم بنسبة 15% باستخدام المعادلة:
$$\text{discount} = \text{price} * 0.15$$
 - إذا كان السعر الإجمالي أقل من 500:
 - لا يوجد خصم:
$$\text{discount} = 0$$
4. حساب السعر النهائي بعد تطبيق الخصم باستخدام المعادلة:
$$\text{final_cost} = \text{price} - (\text{discount})$$
5. طباعة السعر النهائي.
6. نهاية البرنامج.

خريطة التدفق FlowChart :





تابع/ أوراق العمل: الشروط Conditions

تابع/ ورقة عمل (11): برنامج حساب الخصومات على المنتجات

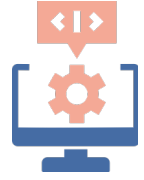
التعليمات البرمجية:

Code

```
1  # Get total price
2  price = float(input('Enter total price: '))
3  if price > 1000:
4      discount = price * 0.25
5  elif price >= 500:
6      discount = price * 0.15
7  else:
8      discount = 0
9  final_cost = price - discount
10 print("Final Cost =", final_cost)
```

Output

```
Enter total price: 1500
Final Cost = 1125.0
```



تابع/ أوراق العمل: الشروط Conditions

ورقة عمل (12): برنامج تصميم آلة حاسبة

المشكلة: استخدام المعاملات الحسابية (+ , - , * , /) في تنفيذ العمليات الحسابية (الجمع، الطرح، الضرب، القسمة).

المدخلات:

- استقبال من المستخدم العدد الأول ثم العدد الثاني.
- استقبال من المستخدم المعامل الحسابي المطلوب.

المعالجة:

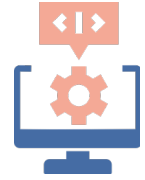
استخدام الشروط لتنفيذ العملية الحسابية المطلوبة.

المخرجات:

- طباعة ناتج العملية الحسابية المطلوبة.
- طباعة "It cannot be divided by zero" في حال كانت العملية الحسابية / والعدد الثاني مساوياً (صفر).
- طباعة "Unsupported operator" في حال ادخال رمز مختلف عن المعاملات الحسابية (+ , - , * , /).

الخوارزمية:

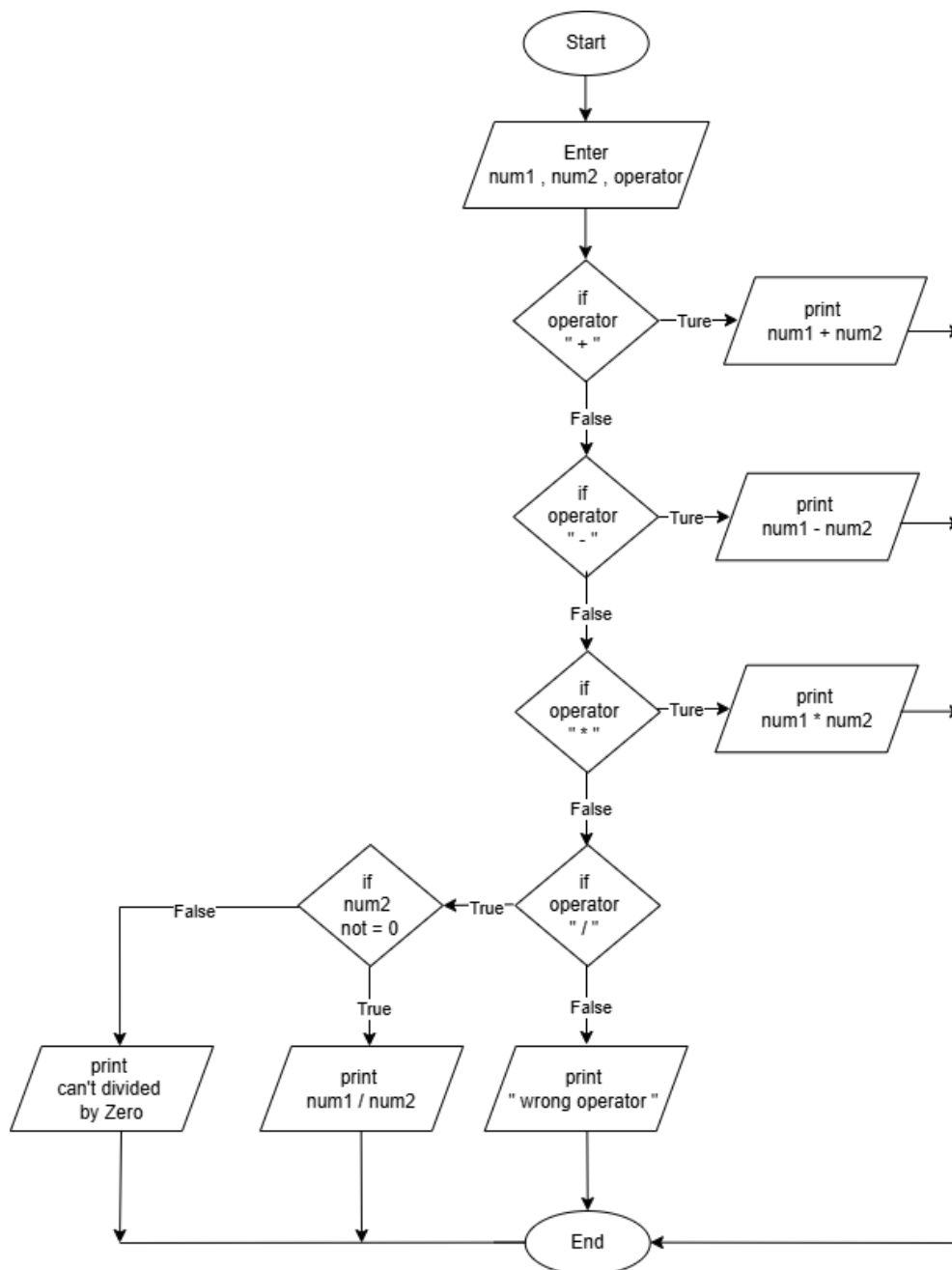
1. بداية البرنامج.
2. استقبال من المستخدم العدد الأول، ثم العدد الثاني.
3. استقبال من المستخدم المعامل الحسابي المطلوب.
4. استخدام الشروط **Conditions**:
 - إذا كان المعامل المُدخل +، يجمع العددين الأول والثاني.
 - إذا كان المعامل المُدخل -، يطرح العدد الثاني من العدد الأول.
 - إذا كان المعامل المُدخل *، يضرب العددين الأول والثاني.
 - إذا كان المعامل المُدخل /، يقسم العدد الأول من العدد الثاني.(التأكد من العدد الثاني لا يساوي صفر، وطباعة رسالة تفيد ذلك).
5. طباعة ناتج العملية الحسابية.
6. نهاية البرنامج.

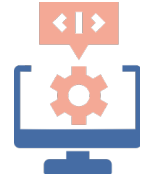


تابع/ أوراق العمل: الشروط Conditions

تابع/ ورقة عمل (12): برنامج تصميم آلة حاسبة

خريطة التدفق FlowChart :





تابع/ أوراق العمل: الشروط Conditions

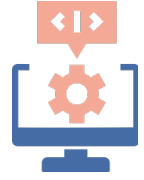
التعليمات البرمجية:

Code

```
1  #Conditions and math. operator
2  num1 = float(input('Enter First Number: '))
3  num2 = float(input('Enter Second Number: '))
4  operator = input('Enter Math. Operator(+, -, *, /): ')
5  if operator == '+':
6      print(num1+ num2)
7  elif operator == '-':
8      print (num1-num2)
9  elif operator == '*':
10     print(num1 * num2)
11  elif operator == '/':
12     if num2 != 0:
13         print(num1/num2)
14     else:
15         print('It cannot be divided by zero')
16  else:
17     print("Unsupported operator")
```

Output

```
Enter First Number: 100
Enter Second Number: 2
Enter Math. Operator(+, -, *, /): /
50.0
```

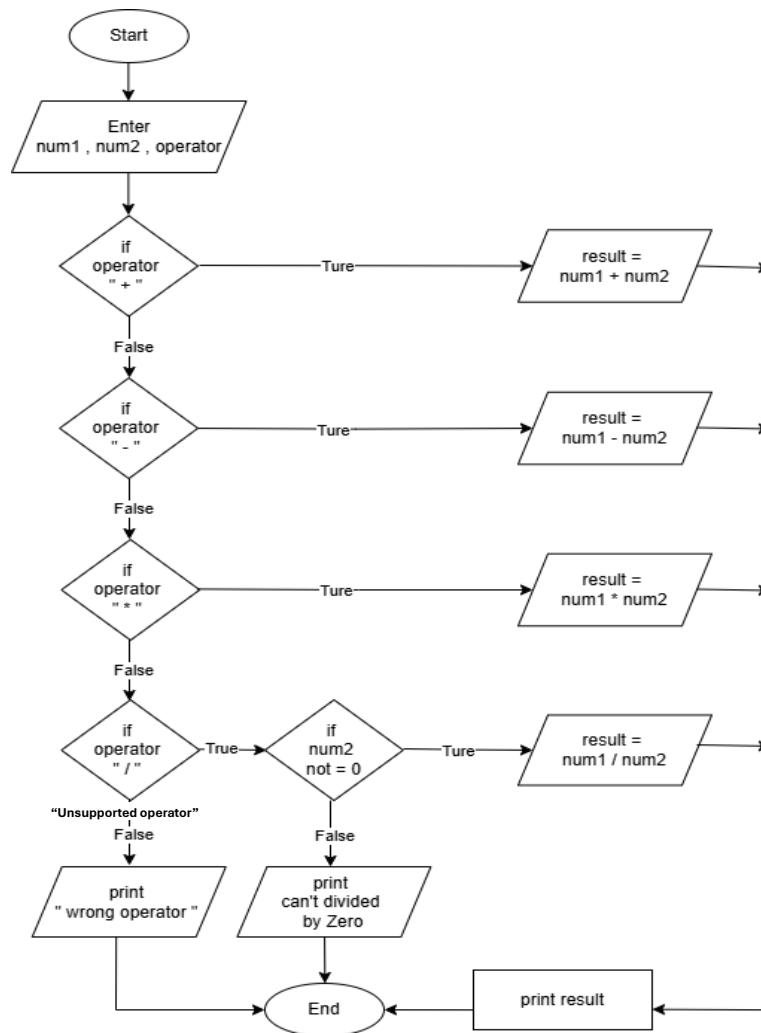


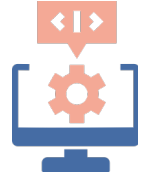
تابع/ أوراق العمل: الشروط Conditions

حل آخر / ورقة عمل (12): برنامج تصميم آلة حاسبة

- إضافة جزء للمعالجة (اجراء العملية الحسابية).
- تليها خطوة الطباعة للنتيجة العملية.
- في المثال التالي إضافة متغير result يتم تخزين ناتج العملية الحسابية.

خريطة التدفق FlowChart :





تابع/ أوراق العمل: الشروط Conditions

حل آخر / ورقة عمل (12): برنامج تصميم آلة حاسبة

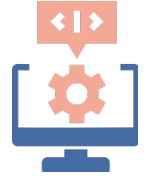
التعليمات البرمجية:

Code

```
1 #Conditions and math. operator
2 num1 = float(input('Enter First Number: '))
3 num2 = float(input('Enter Second Number: '))
4 operator = input('Enter Math. Operator(+, -, *, /): ')
5 if operator == '+':
6     result = num1+ num2
7     print('result: ',result)
8 elif operator == '-':
9     result = num1-num2
10    print('result: ', result)
11 elif operator == '*':
12    result = num1 * num2
13    print('result: ',result)
14 elif operator == '/':
15     if num2 != 0:
16         result = num1/num2
17         print('result: ',result)
18     else:
19         print('It cannot be divided by zero')
20 else:
21     print("Unsupported operator")
```

Output

```
Enter First Number: 100
Enter Second Number: 2
Enter Math. Operator(+, -, *, /): /
result: 50.0
```



مثال إضافي لشرح الدوال الشرطية: الشروط Condition

مثال إضافي: حالة الطقس

في هذا المثال، يطلب من المستخدم إدخال قيمة لـ `weather`، ثم استخدام الدوال الشرطية `if`, `else` لتحديد الرسالة المناسبة بناءً على القيمة المدخلة.

المشكلة:

كتابة برنامج بسيط: إذا كانت السماء تمطر عندها سأستخدم المظلة وإلا سوف أرتدي قبعة.

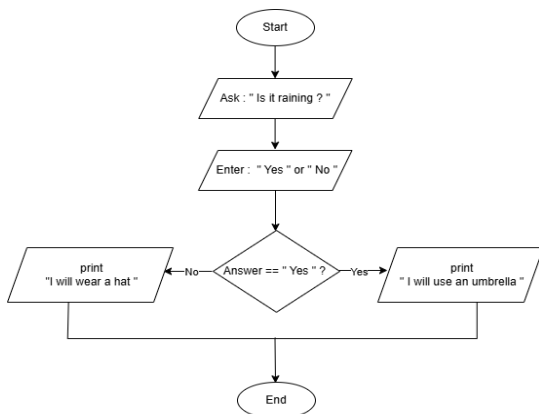
المدخلات: المتغير `weather` و قيمته `yes` أو `no`.

المعالجة: اتخاذ قرار في حال كان `weather` مساوياً `yes` أو `no`.

المخرجات:

- طباعة "I will use an umbrella." إذا كان `weather` مساوياً `yes`.
- طباعة "I will wear a hat." إذا كان `weather` مساوياً `no`.

خريطة التدفق FlowChart :



الخوارزمية :

1. بداية البرنامج.
2. سؤال المستخدم: (is it raining? Please answer yes or no)
3. قراءة الإجابة:
 - إذا كانت الإجابة نعم (yes): اطبع (I will use an umbrella.)
 - إذا كانت الإجابة لا (no): اطبع (I will wear a hat.)
4. نهاية البرنامج.

Code

```

1  برنامج يحدد استخدام المظلة أو القبة بناءً على حالة الطقس
2  print(" is it raining ? please answer yes or no ?")
3  weather = input()
4  if weather == "yes":
5      print("I will use an umbrella.")
6  else:
7      print("I will wear a hat.")
8

```

Output

```

is it raining ? please answer yes or no ?
no
I will wear a hat.

```



الدرس الخامس: التكرار Loop

تُعد الحلقات التكرارية أدوات قوية وفعّالة في البرمجة، وتتيح لك تكرار تنفيذ مجموعة من التعليمات بسهولة، باستخدام `while`، `for` يمكنك تنفيذ التكرار على مجموعة متنوعة من التعليمات، واستخدام الحلقات المتداخلة تزيد من تمكين عمليات متعددة المستويات، ودوال التحكم مثل `break` - `continue` تضيف مزيدًا من التحكم والديناميكية لكيفية تنفيذ التكرار.



خطة الدروس: التكرار Loop.

 Loop1	حصة دراسية	الحلقات التكرارية (while loop) — (while ... else)	
 Loop2	حصة دراسية	الحلقة التكرارية (for loop).	
 Loop3	حصة دراسية	الحلقات المتداخلة (Nested Loops).	
 Loop4	حصة دراسية	تابع: الحلقات المتداخلة (Nested Loops) • دالة (break). • دالة (continue).	





نواتج التعلم: التكرار Loop.

- 1 توضيح مفهوم التكرار وأهميته في تقليل التكرار اليدوي للأوامر داخل البرنامج.
- 2 استخدام الحلقة التكرارية `while` لتنفيذ أوامر طالما الشروط متحققة.
- 3 تطبيق الحلقة التكرارية `for` للتكرار على عناصر مثل السلاسل النصية والنطاقات الرقمية.
- 4 تنفيذ تكرارات متداخلة `Nested Loops` لحل مشكلات معقدة تتطلب أكثر من مستوى من التكرار.
- 5 استخدام التعليمة البرمجية `break` للخروج من الحلقة التكرارية عند تحقق شرط معين.
- 6 توظيف التعليمة `continue` لتخطي تكرار معين دون إنهاء الحلقة بالكامل.
- 7 شرح واستخدام البنية العامة `while...else` لفهم كيفية تنفيذ التعليمات بعد انتهاء التكرار.



مصطلحات حاسوبية: التكرار Loop



الكائن Object



التكرار Loop



الحلقة التكرارية While



الحلقة التكرارية For



تابع/ مصطلحات حاسوبية: التكرار Loop



التكرارات المتداخلة Nested Loops



التكرارات اللانهائية Infinite Loop



قابل للتكرار Iterable



القيمة الابتدائية Initial Value



التعليمة البرمجية continue



التعليمة البرمجية break



الدالة range



استراتيجيات تعليمية مقترحة: التكرار Loop

استخدام استراتيجية المناقشة والحوار: للتعرف على مفهوم التكرار (while)، يمكن مناقشة النقاط التالية بعد استعراض جملة شرطية بسيطة مثل:

```
1 x=1
2 if x < 5:
3     print(x)
4 print("program ended")
```

- ما هو ناتج التعليمات البرمجية السابقة؟
- ما هو ناتج التعليمات البرمجية إذا كانت قيمة المتغير x تساوي 7؟
- كم عدد المرات الذي ستنفذ التعليمة البرمجية (print(x))؟
- ماذا لو استبدلنا (if) بـ (while) ماذا سيحدث؟
 - شرح وظيفة while
- لماذا لم يتم طباعة الرقم (1) بشكل مستمر دون توقف؟
 - الإشارة إلى أن الشرط يتحقق دائما.
- كيف نتغلب على هذه المشكلة؟
 - أن شرط التكرار ينتهي في مرحلة ما، مثل زيادة مقدار المتغير x بمقدار 1، فينهي تحقق شرط التكرار عندما يكون قيمة المتغير 5.
 - استخدام التعليمة break.

استخدام استراتيجية المناقشة والحوار: للتعرف على مفهوم التكرار (for)، حيث يمكن استخدام كشف حضور الطلبة كمثال عملي، ناقش مع الطلبة ما هي الخطوات المتبعة لأخذ الغياب اليومي؟

- احضار دفتر الغياب الخاص في الفصل.
 - قراءة الأسماء بالتسلسل.
 - انتظار الرد من عبارة تبين وجوده في الفصل.
 - احتساب المتعلم غياب في حين عدم وجوده.
- يهدف التمرير بيان الفرق بين عملية التكرار (while) و (for) حيث أن التكرار (while) يستمر باستمرار تحقق الشرط، أما التكرار (for) يستخدم مجموعة من العناصر (مثل أسماء المتعلمين في كشف الغياب) لإجراء عملية التكرار.



استراتيجيات تعليمية مقترحة: التكرار Loop

استخدام استراتيجية المناقشة والحوار (لتعزيز مفهوم المتغيرات واستخدامها مع التكرار): مناقشة كيف يتم عمل البرامج من منظور برمجي للتعرف على أساسيات البرمجة الرئيسية، فيمكن القول:

- أن البرامج (عادةً) تتكون من حلقات تكرارية.
مثال: عند تشغيل أي برنامج مثل محرر الأوامر (PyCharm) يضل البرنامج في حلقة تكرارية مستمرة لكي تصل الواجهة ظاهرة يمكن التفاعل معها.
- ولليانتقال من حلقة تكرارية إلى أخرى نحتاج إلى شرط.
- ولتكوين شرط (عادةً) نقارن بين متغير وقيمة ما باستخدام أحد علامات المقارنة.
- ولليإعلان عن متغير يتوجب علينا معرفة أنواع المتغيرات.

تهدف الاستراتيجية التعليمية لاستعراض الخطوات التي سيتبعها المتعلم لتعلم أساسيات البرمجة وهي التعرف على:

- أنواع المتغيرات.
- كيفية الإعلان عن المتغيرات.
- كيفية إنشاء الشروط.
- كيفية إنشاء الحلقات التكرار.



حل أوراق العمل : التكرار Loop

ورقة عمل (13): برنامج تخمين عدد

المشكلة: إدخال أعداد صحيحة، والتأكد من مدى مطابقتها للعدد المُعرف في البرنامج.

المدخلات: يستقبل من المستخدم عدد (player_number).

المعالجة:

1. الإعلان عن عدد (correct_number).

2. مقارنة العدد المدخل (player_number) بالعدد الصحيح (correct_number) المخزن في البرنامج وتكرار المقارنة

حيث يتوقف عند تخمين المستخدم للعدد الصحيح.

المخرجات: طباعة العبارة المناسبة لكل حالة بعد المقارنة.

الخوارزمية:

1. بداية البرنامج.
2. الاعلان عن متغير يساوي العدد المطلوب تخمينه.
3. استقبال عدد صحيح من المستخدم.
4. مطابقة العدد المدخل مع العدد المحدد في الحالات التالية:
 - إذا كان أكبر يطبع رسالة العدد أكبر ويستمر المستخدم في عملية الادخال.
 - إذا كان أصغر يطبع رسالة العدد أصغر ويستمر المستخدم في عملية الادخال.
 - طباعة رسالة العدد المدخل مطابق للعدد المعرف وينتهي البرنامج.
5. نهاية البرنامج.

التعليمات البرمجية :

Code

```
1 correct_number = 8
2 player_number = 0
3 while player_number != correct_number:                # looping
4     player_number = int (input('Guess what is the number in my mind: '))
5     if player_number > correct_number:
6         print('The player number is higher.')
7     elif player_number < correct_number:                #if the number less
8         print('The player number is lower.')            #show message
9     else:
10        print('Correct! well done.')
11        exit()
```



تابع/ حل أوراق العمل: التكرار Loop

تابع/ ورقة عمل (13): برنامج تخمين عدد.

Output

```
Guess what is the number in my mind: 7
The player number is lower.
Guess what is the number in my mind: 9
The player number is higher.
Guess what is the number in my mind: 8
Correct! well done.
```

تطوير البرنامج لورقة عمل (13) :

إثرائي: تطوير البرنامج السابق (ورقة عمل 13) - لتوليد رقم عشوائي:

Code

```
1  # لتوليد رقم عشوائي من 1 - 30
2  import random
3  correct_number = random.randint(1,30)
4  # -----
5  player_number = 0
6  while player_number!= correct_number:
7      player_number = int(input("Enter a number between 1 and 30: "))
8      if player_number > 30 or player_number <1:
9          print("Please enter a number between 1 and 30")
10         continue
11     if player_number < correct_number:
12         print("The player number is low, Try again.")
13     elif player_number > correct_number:
14         print("The player number is high, Try again.")
15     else:
16         print("Congratulations! You guessed the correct number.")
```

Out put

```
Enter a number between 1 and 30: 15
The player number is high, Try again.
Enter a number between 1 and 30: 10
The player number is low, Try again.
Enter a number between 1 and 30: 13
The player number is low, Try again.
Enter a number between 1 and 30: 14
Congratulations! You guessed the correct number.
```



تابع/ حل أوراق العمل: التكرار Loop

ورقة عمل (14): برنامج جدول الضرب

المشكلة: برنامج يطبع جدول الضرب لعدد محدد.

المدخلات: استقبال العدد المطلوب طباعة جدول الضرب الخاص به.

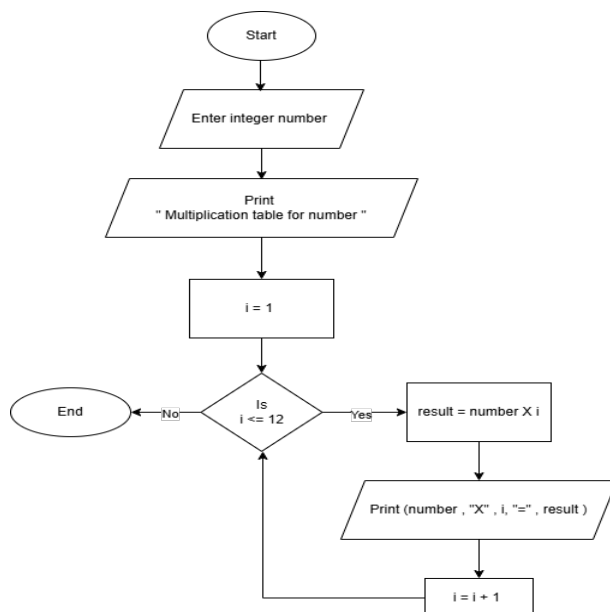
المعالجة: إنشاء حلقة تكرارية لحساب ناتج الضرب باستخدام المعادلة:

الناتج = العدد المدخل * قيمة متغير التكرار.

المخرجات:

- طباعة رسالة (Multiplication table for) ثم قيمة العدد المُدخل.
- طباعة جدول الضرب الخاص بالعدد المدخل.

خريطة التدفق FlowChart :



الخوارزمية :

1. بداية البرنامج.
2. استقبال من المستخدم العدد المطلوب طباعة جدول الضرب الخاص به.
3. طباعة رسالة (Multiplication table for) ثم قيمة العدد السابق.
4. إنشاء حلقة تكرارية لطباعة ضرب العدد المدخل في الأعداد من 1 إلى 12.
 - كتابة التعليمات البرمجية لحساب: الناتج = العدد المدخل * قيمة متغير التكرار.
 - طباعة جدول الضرب.
5. نهاية البرنامج.



تابع/ حل أوراق العمل: التكرار Loop

تابع/ ورقة عمل (14): برنامج جدول الضرب

Code

```
1 num = int(input('Please enter an integer number: '))
2 print('Multiplication table for :', num)
3 for i in range(1, 13):
4     print(num, '*', i, '=', i*num)
```

Output

```
Please enter an integer number: 3
Multiplication table for : 3
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27
3 * 10 = 30
3 * 11 = 33
3 * 12 = 36
```



الدرس السادس:

تصحيح الأخطاء والاستثناءات Debugging and Exceptions

تصحيح الأخطاء (Debugging) والاستثناءات (Exceptions) هما مفهومان أساسيان في البرمجة بلغة بايثون، حيث يمثلان جزءًا حيويًا من عملية تطوير البرامج. تصحيح الأخطاء هي عملية تحديد وإزالة الأخطاء (الأخطاء أو العيوب) من التعليمات البرمجية، بينما الاستثناءات هي أحداث غير متوقعة تحدث أثناء تنفيذ البرنامج وتتسبب في توقف البرنامج عن العمل بشكل طبيعي .

سنتعرف في هذا الدرس على الأخطاء الشائعة مثل: الأخطاء اللغوية (Syntax Errors)، وأخطاء وقت التشغيل (Runtime Errors)، وأخطاء منطقية (Logic Errors).

أيضا سنستخدم عبارات (try) و (except) للتعامل مع الاستثناءات، حيث يتم وضع التعليمات البرمجية التي من المحتمل أن تستثنى داخل كتلة (try)، ويتم كتابة التعليمات البرمجية التي ستتعامل مع الاستثناء داخل كتلة (except). باختصار، تصحيح الأخطاء والاستثناءات يتطلب فهماً جيداً لكيفية عملهما لتجنب المشاكل وضمان تشغيل البرنامج بشكل صحيح.

سنتعلم مهارات تصحيح الأخطاء (Debugging) في تتبع مصدر الخطأ وتحليله ومعالجته بكفاءة، يُعدّ منقّح الأخطاء أداة مفيدة لتعقب الأخطاء حيث يشغّل سطرًا واحد من الشيفرة البرمجية ثم ينتظر حتى تخبره بالمتابعة. يمكنك أن تأخذ الوقت الذي تريده لفحص القيم الموجودة في المتغيرات في أي مرحلة معينة من عمر البرنامج من خلال تشغيل برنامجك مع منقّح الأخطاء باستخدام هذه الطريقة.

يمكنك تشغيل البرنامج مع منقّح أخطاء المحرّر من خلال النقر على زر تنقيح الأخطاء "Debug" في الصف العلوي من الأزرار بجانب زر التشغيل "Run".

يضيف وضع تنقيح الأخطاء أيضًا أزرارًا جديدة إلى أعلى المحرّر وهي: زر المتابعة "Continue" وزر "Step Over" وزر "Step In" وزر "Step Out"، ويوجد زر التوقف "Stop" أيضًا .



خطة الدروس: تصحيح الأخطاء و الاستثناءات

	حصة دراسية	مفهوم الأخطاء (Errors) والاستثناءات (Exceptions) • أنواع الأخطاء الشائعة Errors . - أخطاء نحوية (Syntax Errors). - أخطاء وقت التشغيل (Runtime Errors). - أخطاء المنطق (Logic Errors).	
	حصتان دراسيتان	تابع / تصحيح الأخطاء و الاستثناءات • التعامل مع الاستثناءات Exceptions. • التعليمة البرمجية try / except • تصحيح الأخطاء Debugging	



نواتج التعلم: تصحيح الأخطاء والاستثناءات

- 1 شرح مفهوم الأخطاء Errors، والاستثناءات Exceptions في البرامج.
- 2 التمييز بين أنواع الأخطاء الشائعة مثل: أخطاء بناء الجملة Syntax Errors، وأخطاء وقت التشغيل Runtime Errors، والأخطاء المنطقية Logic Errors.
- 3 شرح أهمية التعامل مع الاستثناءات لضمان استقرار البرنامج ومنع توقفه المفاجئ.
- 4 استخدام التعليمة try/except لمعالجة الأخطاء المتوقعة وتقديم حلول بديلة أو رسائل توضيحية.
- 5 تطبيق مهارات تصحيح الأخطاء Debugging في تتبع مصدر الخطأ وتحليله ومعالجته بكفاءة.



مصطلحات حاسوبية: تصحيح الأخطاء والاستثناءات



تصحيح الأخطاء Debugging



الأخطاء Errors



أخطاء وقت التشغيل Runtime Errors



الأخطاء اللغوية Syntax Errors



القسمة على صفر ZeroDivision Error



الأخطاء المنطقية Logic Errors



نوع بيانات غير صحيحة Type Error



متغير غير معرف Name Error



التعليمة البرمجية Try



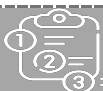
قيمة غير مناسبة لنوع البيانات Value Error



الكتلة البرمجية finally



التعليمة البرمجية except



تنفيذ سطر تلو الآخر Step Over/Into



نقطة توقف BreakPoint



استثناءات شاملة Catch-all Exceptions



استثناءات محددة Specific Exceptions



ملاحظات للمعلم:

الاطلاع على تقنيات تصحيح الأخطاء الشائعة Python وسيتم عرض أبسطهم والتي تناسب مع الفئة العمرية للمتعلمين .

الطباعة باستخدام `print()`

الأسلوب الأبسط ، حيث يمكنك وضع أمر `print()` بعد تعليمات برمجية لرؤية قيم المتغيرات أو التحقق من الخطوات.



نقاط التوقف (Breakpoints)

هي ميزة في PyCharm تتيح لك إيقاف التعليمات البرمجية مؤقتاً عند سطر معين، للتمكن من فحص قيم المتغيرات أو التحقق من التعليمات البرمجية خطوة بخطوة.



قراءة رسائل الخطأ في PyCharm

PyCharm يعرض رسائل خطأ مفصلة تساعدك على معرفة نوع الخطأ وموقعه.



كتابة `comment` قبل التعليمات البرمجية

لشرح الهدف من التعليمات البرمجية أو توضيح الأجزاء (المقاطع) البرمجية .



التعامل مع الأخطاء باستخدام `try/except`

في حالة التوقع حدوث خطأ، يمكن استخدام `try/except` لمنع توقف البرنامج فجأة يساعد على التعامل مع الأخطاء وتوفير رسائل توضيحية للمستخدم.



وغيرها من التقنيات لتصحيح الأخطاء.



تصحيح الأخطاء في بايثون Debugging :

تصحيح الأخطاء في بايثون (Debugging) هو عملية تحديد وإزالة الأخطاء من برنامج بايثون.

خطوات تصحيح الأخطاء في بايثون:

1. تحديد نوع الخطأ: أول خطوة هي تحديد نوع الخطأ في البرنامج.
2. استخدام مصحح الأخطاء مثل:
 - PDB: يعد مصحح الأخطاء المدمج في بايثون (PDB) أداة قوية لتصحيح الأخطاء.
3. تتبع الأخطاء:
 - الخطوة الأولى : مثلاً استخدام PDB لتحديد مكان الخطأ في الكود.
 - الخطوة الثانية : فحص قيم المتغيرات في نقاط مختلفة من الكود لتحديد سبب الخطأ.
 - الخطوة الثالثة : إجراء التعديلات اللازمة على الكود لإصلاح الخطأ.
4. اختبار الأخطاء:
 - الخطوة الأولى : تشغيل الكود بعد إجراء التعديلات.
 - الخطوة الثانية : تأكيد أن الكود يعمل بشكل صحيح وأنه لا توجد أخطاء أخرى.

نصائح لتصحيح الأخطاء:

- اجعل الكود بسيطاً : كلما كان الكود أبسط، كان من الأسهل تحديد الأخطاء.
 - استخدم التعليقات : التعليقات في الكود تساعد على فهمه وتسهيل عملية التصحيح.
 - اختبر الكود بشكل منتظم: لا تنتظر حتى يكتمل الكود لتصحيحه، اختبره بشكل دوري.
 - استخدم المصححات: استخدم PDB أو IDE لتصحيح الأخطاء، فهي أداة قوية.
 - ابحث عن الأخطاء الشائعة: تعرف على الأخطاء الشائعة في بايثون وحاول تجنبها.
- باستخدام هذه الأدوات والتقنيات، يمكنك أن تصبح مصحح أخطاء فعالاً في بايثون وتضمن تشغيل برامجك بسلاسة.



أوراق العمل : تصحيح الأخطاء والاستثناءات

ورقة عمل (15): حساب قيمة فاتورة المشتريات بعد الخصم.

المشكلة: حساب المستخدم السعر بعد تطبيق خصم الفاتورة، من خلال:

- إدخال قيمة الفاتورة (أكبر من صفر).
- إدخال نسبة خصم (تتراوح بين 0 و100).

المدخلات: السعر الأصلي (price) ونسبة الخصم (discount).

المعالجة: حساب السعر النهائي باستخدام المعادلة:

$$\text{net_price} = \text{price} - (\text{price} * \text{discount} / 100)$$

المخرجات:

- طباعة السعر بعد الخصم (net_price).
- طباعة (Error: Please enter valid numeric values) في حال وجود خطأ من النوع (ValueError).
- طباعة (Program has ended).

الخوارزمية:

1. بداية البرنامج.
2. استخدام كتلة try لمحاولة تنفيذ الخطوات التالية:
 - a. طلب من المستخدم إدخال price (السعر الأصلي).
 - b. طلب من المستخدم إدخال discount (نسبة الخصم).
 - c. حساب السعر النهائي باستخدام:

$$\text{net_price} = \text{price} - (\text{price} * \text{discount} / 100)$$
 - d. طباعة "Net price after discount is: net_price"
3. استخدام كتلة except بحيث إذا حدث خطأ في نوع البيانات (ValueError):

طباعة: "Error: Please enter valid numeric values"
4. في جميع الأحوال، طباعة الرسالة النهائية:

"Program has ended" داخل finally.
5. نهاية البرنامج.



تابع / أوراق العمل: تصحيح الأخطاء و الاستثناءات

تابع/ ورقة عمل (15): حساب قيمة فاتورة المشتريات بعد الخصم.

التعليمات البرمجية :

Code

```
1  try:
2      price = float(input('Enter the original price (KD): '))
3      discount = float(input('Enter discount percentage (%): '))
4      if price > 0:
5          if discount > 0 and discount <= 100:
6              net_price = price - (price * discount/100)
7              print(f'Net price after discount is: {net_price}')
8
9  except ValueError:
10     print('Error: Please enter valid numeric values')
11
12 finally:
13     print('Program has ended')
```

Output

```
Enter the original price (KD): -1
Enter discount percentage (%): 10
Program has ended
```



Code

```
1  try:
2      price = float(input('Enter the original price (KD): '))
3      discount = float(input('Enter discount percentage (%): '))
4      if price <= 0:
5          print('Error: Price must be greater than zero')
6
7      elif discount < 0 or discount > 100:
8          print('Error: Discount must be between 0% and 100%')
9
10     else:
11         net_price = price - (price * discount / 100)
12         print(f'Net price after discount is: {net_price}')
13
14 except ValueError:
15     print('Error: Please enter valid numeric values')
16
17 finally:
18     print('Program has ended')
```

Output

```
Enter the original price (KD): 600g
Error: Please enter valid numeric values
Program has ended
```



تابع / أوراق العمل: تصحيح الأخطاء و الاستثناءات

ورقة عمل (16): مؤشر كتلة الجسم BMI (Body mass index)

المشكلة: اكتشاف الأخطاء وتجاوزها (استكمال البرنامج ليعمل بصورة صحيحة).

المدخلات: الوزن بالكيلوجرام (weight) والطول بالمتر (height).

المعالجة:

- حساب مؤشر كتلة الجسم BMI باستخدام المعادلة:

$$\text{قانون كتلة الجسم: (BMI)} = \frac{\text{الوزن بالكيلوجرام}}{\text{الطول بالمتر المربع}}$$

- تصنيف الحالة الصحية حسب نتيجة BMI :

- إذا كان BMI أقل من 18.5.
- إذا كان BMI بين 18.5 و 24.9.
- إذا كان BMI بين 25 و 29.9.
- إذا كان BMI غير ذلك.

- معالجة الأخطاء:

- إذا أدخل المستخدم نص أو قيمة غير رقمية (ValueError).
- إذا أدخل الطول 0 = (ZeroDivisionError).

المخرجات:

- طباعة قيمة مؤشر كتلة الجسم (BMI) بدقة خانتين عشريتين.
- وصف الحالة الصحية بناءً على قيمة BMI بطباعة الرسالة المناسبة.
- طباعة رسائل الخطأ في حال وجود خطأ في الإدخال (ZeroDivisionError , ValueError).

الخوارزمية:

1. بداية البرنامج.
2. بداية التكرار اللانهائي وتنفيذ التالي:
 - استخدام كتلة try لمحاولة تنفيذ الخطوات التالية:
 - طلب من المستخدم إدخال الوزن بالكيلوجرام وحفظه كعدد عشري (float).
 - طلب من المستخدم إدخال الطول بالمتر وحفظه كعدد عشري (float).
 - حساب مؤشر كتلة الجسم BMI باستخدام المعادلة:

$$\text{قانون كتلة الجسم: (BMI)} = \frac{\text{الوزن بالكيلوجرام}}{\text{الطول بالمتر المربع}}$$
 - طباعة قيمة BMI بتنسيق عشري من خانتين.
 - تحديد الفئة الصحية بناءً على قيمة BMI :
 - (a) إذا كان BMI أقل من 18.5، طباعة "You are underweight."
 - (b) إذا كان BMI بين 18.5 و 24.9، طباعة "You have a normal weight."



تابع / أوراق العمل: تصحيح الأخطاء و الاستثناءات

تابع / ورقة عمل (16): مؤشر كتلة الجسم BMI (Body mass index)

- (c) إذا كان BMI بين 25 و 29.9، طباعة "You are overweight."
- (d) إذا كان BMI غير ذلك طباعة "You are obese."
- الاستمرار بالتكرار عند الضغط على مفتاح Enter.
- الخروج من التكرار اللانهائي عند كتابة الحرف Q.
- استخدام كتلة except بحيث إذا أدخل المستخدم قيمة غير رقمية (ValueError):
(a) طباعة رسالة: "Error: Please enter valid numeric values for weight and height."
- استخدام كتلة except بحيث إذا حدث خطأ وأدخل المستخدم الطول صفراً (لا يمكن القسمة على صفر) (ZeroDivisionError):
(a) طباعة رسالة: "Error: You can't divide by zero!"
- 3. الرجوع إلى الخطوة 2.
- 4. نهاية البرنامج.

التعليمات البرمجية :

Code

```
1 # Program to calculate BMI with error handling
2 while True:
3     try:
4         # Get user input: weight and height
5         weight_kg = float(input('Enter your weight in kilograms: '))
6         height_m = float(input('Enter your height in meters: '))
7         # Calculate BMI
8         bmi_value = weight_kg / (height_m ** 2)
9         # Display the BMI
10        print(f'Your BMI is: {bmi_value:.2f}')
11        # Determine BMI category
12        if bmi_value < 18.5:
13            print('You are underweight.')
14        elif bmi_value >= 18.5 and bmi_value < 25:
15            print('You have a normal weight.')
16        elif bmi_value >= 25 and bmi_value < 30:
17            print('You are overweight.')
18        else:
19            print('You are obese.')
20        stop_program = input('Press Enter to continue or Q to quit: ').capitalize()
21        if stop_program == 'Q':
22            break
23    except ValueError:
24        print('Error: Please enter valid numeric values for weight and height.')
25    except ZeroDivisionError:
26        print('Error: You cannot divide by zero!')
```



Output

```
Enter your weight in kilograms: 80
Enter your height in meters: 1.80
Your BMI is: 24.69
You have a normal weight.
Press Enter to continue or Q to quit.
Enter your weight in kilograms: 90
Enter your height in meters: 1.50
Your BMI is: 40.00
You are obese.
Press Enter to continue or Q to quit.
Enter your weight in kilograms: 60
Enter your height in meters: 0
Error: You cannot divide by zero!
Enter your weight in kilograms: 60
Enter your height in meters: 1.40
Your BMI is: 30.61
You are obese.
Press Enter to continue or Q to quit.q
```

الوحدة الثالثة: المنتجات الرقمية



المنتجات الرقمية : المشروع

تختلف وحدة (المنتجات الرقمية) عن باقي وحدات الكتاب كونها لا تعتمد على تنفيذ أوراق عمل محددة ومعدة مسبقاً للمتعلم، وإنما تعتمد على إبداع المتعلم سواء كان منفرداً أو ضمن مجموعة في توظيف ما تم تعلمه من معارف ومهارات حاسوبية في إنتاج مشروع بسيط يخدم موضوع معين أو يقدم خدمة معينة أو يعرض معلومة بشكل مختلف. يقدم المتعلم في وحدة المنتجات الرقمية مشروعاً من خلال ما تعلمه من مهارات ومعارف في لغة البايثون والمعلومات التي تم تطبيقها من خلال الدروس وتطبيق أوراق العمل والأمثلة خلال الفصل الدراسي.



Project



نواتج التعلم: المنتجات الرقمية

1. توظيف المهارات البرمجية المكتسبة في Python بشكل فعال لإنتاج برامج متنوعة.
2. تنمية القدرة على الابتكار والإبداع من خلال تحليل المشكلات وتصميم حلول برمجية مناسبة.
3. تطبيق المهارات البرمجية في تصميم وتطوير برامج تعالج احتياجات من واقع الحياة اليومية.
4. المشاركة بفاعلية ضمن فرق العمل البرمجية، وإظهار روح التعاون وتبادل المعرفة.
5. عرض الأفكار البرمجية أمام الآخرين، ومناقشة الحلول المختلفة مع تقبل وجهات النظر المتنوعة.
6. التفاعل مع التغذية الراجعة لتحسين جودة المشروع، وتطوير الأداء البرمجي.



مراحل إعداد المشروع:

1- إعداد فريق العمل

تحديد فريق العمل لإعداد المشروع من 4 إلى 5 متعلمين (أو حسب ما يراه المعلم مناسباً) في الأسابيع الأولى من الفصل الدراسي بعد استعراض فكرة عامة عن طبيعة المنتجات الرقمية.



2- تحديد الفكرة

تحديد موضوع المشروع والهدف منه ضمن ضوابط وشروط قبول فكرة المنتج الرقمي ومن أهمها توظيف أحد تقنيات الذكاء الاصطناعي في البرنامج ولا يقبل المنتج الرقمي ويعتبر مرفوضاً من دونه، والحرص على بساطة المنتج الرقمي لتجنب أي تعقيدات واستغراق وقت أكثر.



3- إعداد خطة الحل (الخوارزمية)

إعداد خوارزمية الحل لتحقيق فكرة المشروع ومراجعتها من قبل المعلم حيث يقتصر دور المعلم على إرشاد المتعلمين نحو الطريق الصحيح.



4- تصميم خريطة التدفق

تمثيل خوارزمية الحل بخريطة تدفق (Flowchart).



5- كتابة التعليمات البرمجية

تحويل خريطة التدفق الى مجموعة من التعليمات البرمجية باستخدام لغة بايثون





تابع - مراحل اعداد المشروع:

6- اختبار البرنامج ، تصحيح البرنامج إن وجد.

تشغيل البرنامج وإدخال بيانات ومقارنة النتائج لاختبار صحة وعمل البرنامج بصورة سليمة، وتصحيح الأخطاء ان وجدت.



7- عرض المشروع على الزملاء و معلم الحاسوب

مناقشة المشروع مع زملائه بعد عرض تقديمي يقوم به الفريق في الحصة المخصصة لتعزيز الثقة في النفس والتدرب على الإلقاء ومواجهة الجمهور وتقبل النقد من الطرف الآخر.





المهارات المقترحة للمشروع:

من المهم أن يتضمن مشروع المتعلم معظم المهارات التي اكتسبها خلال الفصل الدراسي ومنها:

م	البند	مثال
1	انشاء مشروع جديد	File ____ > new project
2	انشاء ملف بايثون جديد	New ---- → python file
3	الإعلان عن المتغيرات (نصية , عددية , منطقية)	str () , int() , float() , bool()
4	دالة الإخراج	Print ()
5	استخدام العمليات الحسابية	**, %, //, /, *, -, +
6	دالة استقبال البيانات	input()
7	استخدام العمليات المنطقية	and, or
8	استخدام العمليات الشرطية	if, elif, else
9	استخدام الدوال التكرارية	while , for
10	انشاء السلاسل النصية	
11	فهرسة السلاسل النصية	Indexing []
12	تقسيم السلاسل النصية slicing	
13	جمع السلاسل النصية concatenation	+
14	تكرار عناصر السلسلة النصية	*
15	استخدام دوال السلاسل النصية	- capitalize() - - upper() - - lower() - - find() - - replace() - - f-string - - int() - - float() - - center() -



متابعة مراحل إعداد المشروع:

يبدأ المتعلم العمل بالمشروع بدءاً من الأسبوع السابع، ويُستكمل تدريجياً خلال الأسابيع اللاحقة ويمكن للمعلم متابعة مشاريع المتعلمين من خلال الجدول التالي:

		متابعة مراحل اعداد المشروع						
		الأسبوع 12	الأسبوع 11	الأسبوع 10	الأسبوع 9	الأسبوع 8	الأسبوع 7	
ملاحظات	التقييم النهائي	عرض ومناقشة المشروع	اختبار البرنامج وتصحيح الأخطاء	كتابة البرنامج	اعداد خريطة التدفق	اعداد خطة الحل (الخوارزمية)	تحديد الفكرة	اسم المشروع وأسماء أعضاء المشروع
								مشروع:
								-1
								-2
								-3
								-4
								مشروع:
								-1
								-2
								-3
								-4



أمور يجب مراعاتها في وحدة المنتجات الرقمية:



1. تقديم فكرة عن فصل المنتجات الرقمية (وحدة المشاريع) قبل البدء فيها بوقت كافٍ حتى يستعد المتعلمون نفسياً وذهنياً لها مع بداية الفصل الدراسي.



2. الحرص على تقديم نماذج من مشاريع جيدة تم إنجازها مسبقاً أو إعدادها المعلم أو المتعلمون حتى يتضح مفهوم المشروع في أذهان المتعلمين.



3. الحرص على متابعة فرق العمل والتأكيد على العمل التعاوني وقيام كل متعلم بدوره، وتكثيف المتابعة للفرق التي تحتاج وحل المشاكل الفنية التي تواجه المتعلمين أولاً بأول.



4. ترك الحرية للمتعلمين في اختيار أعضاء الفريق للمشروع مع الالتزام بعدد الأعضاء بحيث لا يقل عن اثنين ولا يزيد عن ستة أعضاء مع تحديد القائد ليكون مسئولاً عن التواصل مع المعلم وتنسيق الفريق.



5. ضرورة تجميع مشاريع المتعلمين على وسيط تخزين مناسب وتسليمها لرئيس القسم مع الدرجات الخاصة بها.

6. لتسهيل عملية اجتماع الفرق والعمل الجماعي، يمكن استخدام قنوات الفصل في برنامج Microsoft Teams للاجتماعات وتنسيق العمل.

تحضير الدروس

05



تحضير الدروس:

أهمية تحضير الدرس للمعلم

1	يساعد المعلم على تنظيم أفكاره وترتيبها، فيتوقع المواقف التعليمية التي قد تظهر له.
2	يعين المعلم على اختيار الأنشطة والاستراتيجيات المناسبة للدرس.
3	يعتبر إعداد الدرس سجلاً مكتوباً لمحتوى التعلم يرجع إليه المعلم إذا نسي شيئاً أثناء الحصة.
4	يستطيع المعلم من خلال التحضير اليومي أن يعرف ماذا شرح للمتعلم، من مفاهيم ومعلومات، وماذا تبقى له من مفاهيم لم يتمكن من شرحها وتم تأجيلها للحصة القادمة بسبب ظرف طارئ.
5	يعد إعداد الدرس مهماً من الناحية التربوية، إذ يعتمد عليه مدير المدرسة أو الموجه الفني لمتابعة المعلم أثناء زيارة الحصة.

سجل التحضير

يجب أن يتضمن السجل الصفحات التالية:

1	بيانات المعلم الأساسية. (الاسم الرباعي، الرقم المدني، الجنسية، رقم الملف، تاريخ التعيين، مركز العمل السابق، المؤهل).
2	توقيت اليوم الدراسي.
3	الجدول الدراسي للمعلم والمواد التي يقوم بتدريسها، ثم الأعمال المسندة إليه سواء أنشطة أو مهام إدارية.
4	خطة توزيع المقرر للفصل الدراسي الحالي.
5	الأهداف التربوية. (الهدف الشامل للتربية بالكويت، أهداف المرحلة التعليمية، أهداف تدريس الحاسوب العامة)
6	التحضير اليومي للدرس حسب خطة توزيع المقرر.
7	يمكن أن يضم سجل متابعة المتعلمين اليومي لسجل التحضير أو أن يرفق للحصة بسجل مستقل.

يصاغ بحسب خطة المقرر الدراسي			
الوحدة	اليوم	التاريخ	الفصل
يصاغ من الكتاب المدرسي			

يتم تسجيلها لكل حصة دراسية

مصادر التعلم والوسائل التعليمية	القيمة التربوية
تسجل المصادر التي يستخدمها أثناء عرض الدرس لإثراء الموقف التعليمي (مواد، أدوات رقمية) لا يوجد عدد معين للمصادر التي يمكن استخدامها في الدرس الواحد فهذا يتوقف على طبيعة الدروس	تحديد القيمة التربوية حسب الجدول الزمني المقترح مع وصف زمني لكيفية تفعيلها خلال الحصة بما يتلاءم مع عرض المفاهيم الدرس

الزمن	النشاط الاستهلاكي	المقدمة والتمهيد
5-3 دقائق	تهيئة عقول المتعلمين للمعلومات الجديدة وتشويقهم لها وتختلف حسب موضوع الدرس، كأمثلة للمقدمة: - عرض الناتج النهائي للوحدة ليكون المتعلم فكرة عن المفاهيم التي ستعرض في الدروس القادمة - مراجعة جزء سبق دراسته له علاقة بالدرس - عرض فيديو متعلق بموضوع الدرس او استعراض مواقع وصفحات على النت. *يراعى في المقدمة وضوحها ومناسبتها للفئة العمرية للمتعلمين وارتباطها بالدرس وعدم الإطالة بالمقدمة والتأثير على الزمن المخصص لشرح الدرس	

م	نواتج التعلم	محتوى التعلم (الأنشطة والاستراتيجيات)	التقويم	الزمن
	عبارة تتضمن ما الذي يتوقع من المعلم تحقيقه نتيجة للأنشطة التعلم ويعبر عنها على شكل معارف أو مهارات أو اتجاهات ويجب أن تكون واضحة تصف ما الذي يتوقع ان يتعلمه المتعلمين نتيجة احتمال عملية التعلم	محتوى التعلم يجب أن يجب عن السؤال التالي: ما الذي يجب تعلمه؟ ويص الأنشطة والاستراتيجيات المستخدمة مع وصف مختصر للدور المناط بكل من المعلم والتعلم	صياغة لغوية واضحة للأسئلة والتعليمات التي توجه نشاط المتعلم منعا للارتجال وتحسينا للتواصل اللفظي مع المتعلم باستخدام أدوات ووسائل التقويم، وتعكس هذه الأدوات إنجازات المتعلم وقياسها في مواقف حقيقية، فيبدو كمنشآت تعلم يمارس فيه المتعلمين مهارات التفكير العليا وحل المشكلات الحياتية الحقيقية	تقدير الزمن المناسب لتنفيذ النشاط لاستثمار وقت الحصة
	التقويم الختامي	حل ورقة عمل: ص: أو ورقة عمل بديلة من تصميم المعلم بحيث تقيس النواتج بشكل شامل		

اليوم والتاريخ	الفصل	الحصة	التأمل الذاتي		
			الرضا عن تحقق الأهداف	تحديات واجهتني	مقترحات لتحسين الأداء
			  	التعرف على الأسباب والمشكلات ووضع خطط حلولها (من التحديات توقف النت – ضعف أحد الأنشطة لتوصيل المهارة المطلوبة)	يقود المعلم الى تحسين أدائه واجراء التغييرات المناسبة التي تنعكس ايجابياً على التعليم (استبدال نشاط بآخر والتي تساعد على تحسين الأداء



خطوات اعداد الدرس:

- . القراءة الدقيقة في موضوع الدرس من خلال: الكتاب المدرسي، دليل المعلم أو أي من مختلف المراجع.
- . تحديد أهداف الدرس / نواتج التعلم بدقة ووضوح، وصياغتها صياغة صحيحة.
- . الإعداد الذهني وهو رسم الخطة لتحقيق تلك الأهداف/ نواتج التعلم. وقبل أن تبدأ في الكتابة يجب أن تكون فكرة خطة التدريس قد تبلورت في ذهنك.
- . الإعداد الكتابي بعد أن تكون تصورت تصورا كاملا ومتربطاً لطريقة سير الدرس قم بتسجيلها على شكل خطوات واضحة ومحددة، شاملةً عناصر الإعداد، مراعيًا في كل خطوة عامل الوقت وارتباطها بأهداف الدرس.
- . الإعداد الكتابي بعد تكوين تصور كامل ومتربط لطريقة سير الدرس، يتم تسجيله كتابياً على شكل خطوات واضحة ومحددة، تتضمن جميع عناصر الإعداد، مع مراعاة عامل الوقت وارتباط كل خطوة بأهداف الدرس.
- . إعداد متطلبات الدرس، غالباً ما يحتاج المعلم في الشرح لبعض الأدوات، وينبغي عليه الاهتمام بتحضيرها والتأكد من صلاحيتها وإمكانية استخدامها في المكان الذي ستستخدم فيه.

أخطاء شائعة يقع فيها المعلم عند إعداد الدرس

- . الإهمال في كتابة التاريخ أو لفصل واحد فقط.
- . تجاهل استخدام وسيلة تعليمية أثناء الشرح، تم الإشارة إليها في بند مصادر التعلم والوسائل التعليمية.
- . في المقدمة والنشاط الاستهلالي لا يصح كتابة) مراجعة قصيرة للدرس السابق وإنما تحدد بنود المراجعة).
- . عدم ذكر حل التمارين صفحة . . . أو تترك بدون حل.
- . عدم مناسبة أسلوب العرض من حيث عدد النقاط فلا تكون قصيرة جداً ولا طويلة جداً.
- . عدم تحديد دور المعلم أو دور المتعلم بشكل واضح.
- . اختيار أسئلة تقويم بصياغة خاطئة. (الأسئلة المربكة التي تطرح أكثر من فكرة، الأسئلة التي تكون إجابتها بنعم / لا، الأسئلة الغامضة غير محددة الهدف، الأسئلة الموحية بالإجابة).
- . إهمال المعلم لتسجيل ملاحظاته في جزء التأمل الذاتي.

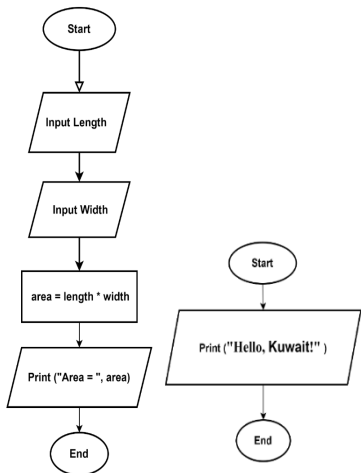
نموذج تحضير درس مادة الحاسوب للصف العاشر

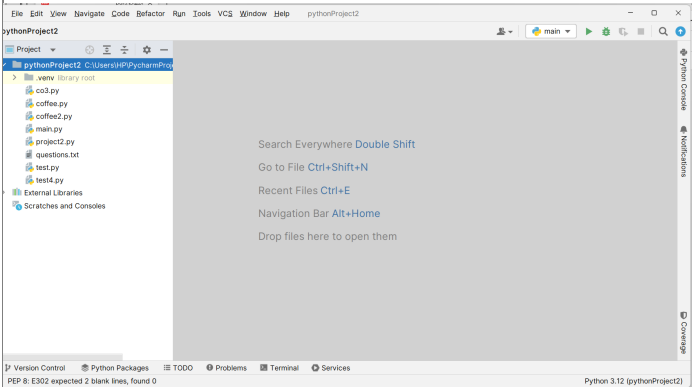
الأمن السيبراني Cyber Security - مدخل الى البرمجة				عنوان الدرس
الأسبوع	الوحدة	اليوم	التاريخ	الفصل
الأول	الأمن السيبراني برمجة Python	الأربعاء	2026-9-18	1/10
		الخميس	2026-9-19	2/10

مصادر التعلم والوسائل التعليمية	القيمة التربوية
السبورة والاقلام — جهاز Data show — بطاقات — سبورة متعلم — فيديو خارجي — عرض تقديمي	توضيح مفهوم الأمن السيبراني ونشر التوعية بمخاطر الهجمات السيبرانية وطرق الوقاية والتخلص منها

المقدمة والتمهيد	النشاط الاستهلالي	الزمن
	- الترحيب بالمتعلمين في بداية العام الدراسي - عرض قوانين المختبر - عرض محتوى المنهج الدراسي والية توزيع الدرجات	5د

م	نواتج التعلم	محتوى التعلم (الأنشطة والاستراتيجيات)	التقويم	الزمن
1	- الفهم الشامل لمبادئ وممارسات أمن المعلومات. - التزود بالمعرفة والمهارات اللازمة لحماية الأنظمة والشبكات والمعلومات ضد الهجمات السيبرانية. - التعريف بمفهوم الأمن السيبراني وأهميته لحماية الأنظمة والشبكات. - التمييز بين مفهومي أمن المعلومات والأمن السيبراني . - تعزيز الوعي الأمني لمخاطر الهجمات السيبرانية عند المتعلمين.	- من خلال استراتيجية سرد قصة: - يعرض قصة هجوم "WannaCry" السيبراني الذي أصاب عشرات الآلاف من الأجهزة في عدد كبير من الدول واستخدام برمجيات خبيثة تسمى "الفدية" وكيفية تأثيره على الأفراد والمؤسسات . - من خلال استراتيجية المناقشة: - عرض المخاطر وتحديد التهديدات الأمنية وكيفية التصدي واتخاذ إجراءات لحماية معلوماتهم. - من خلال العرض التقديمي: - يتم عرض الفيديو التوعوي عن الأمن السيبراني المعد من قبل التوجيه العام	- مناقشة المتعلمين عن طريق السؤال ما هو تأثير هذه البرامج على المجتمع والأفراد؟ - مناقشة المتعلمين عن طريق السؤال كيف نحتمي بياناتنا الشخصية من الاختراق؟ - توزيع بطاقات تحوي أسئلة وعدة عبارات - من خلال ما تم عرضه وضع مفهوم الأمن السيبراني ؟ - مفهوم أمن المعلومات؟ - عرض برنامج أسئلة وأجوبة تفاعلية لقياس فهم المتعلمين للأهمية الحيوية للأمن السيبراني. - عدد أهمية اتخاذ إجراءات وقائية لحماية الأنظمة والمعلومات.	5 5 5

م	نواتج التعلم	محتوى التعلم (الأنشطة والاستراتيجيات)	التقويم	الزمن
2	- مدخل الى البرمجة: مفهوم البرمجة	- من خلال استراتيجية العصف الذهني: - يتم طرح سؤال كيف يمكن للألعاب الفيديو، التطبيقات، وحتى المواقع الإلكترونية أن تعمل وتؤدي وظائفها بشكل دقيق؟ - من خلال استراتيجية التشبيه الواقعي: - طرح مثال يشبه الحاسوب هو طاه في مطبخ - والبرمجة هي كتابة وصفة لطبق معين - كل خطوة في الوصفة هي تعليمات - يجب أن يتبعها الطاهي لتحضير وجبة لذیذة أي انجاز البرنامج النهائي	- أتوقع الإجابة من المتعلمين - كل هذا يتم بفضل البرمجة - سؤال: ما مفهوم البرمجة؟	5
3	- خطوات انشاء برنامج لغة البرمجة python	- من خلال عرض تقديمي: - يتم عرض ومناقشة خطوات انشاء البرنامج (تحديد المشكلة – اعداد الخوارزمية – اعداد خريطة التدفق – كتابة البرنامج – اختبار البرنامج -اصدار البرنامج)	- مناقشة المتعلمين حول المفاهيم التالية : - ماهي الخوارزمية Algorithm ؟ - ما هي خريطة التدفق Flowchart ؟	5
4	- كتابة خوارزمية لحل مشكلة	- من خلال استراتيجية التعلم القائم على المشاريع: - تحديد المشكلة (إيجاد مساحة مستطيل) - استقبال قيمة الطول من المستخدم (بالمتر). - استقبال قيمة العرض من المستخدم (بالمتر). - العملية احسب المساحة باستخدام المعادلة : - المساحة = الطول × العرض - اطبع قيمة المساحة على الشاشة.	- مناقشة المتعلمين: - ما هي المعطيات المتاحة؟ - أطوال اضلاع المستطيل : طول و عرض المستطيل - ما هي العمليات والإجراءات المطلوبة؟ - قانون المساحة المستطيل = الطول * العرض - توزيع المتعلمين على مجموعات ثم توزيع سيورات لتنفيذ التالي: - كتابة الخوارزمية قبل تطبيقها برمجيا	10
5	- تصميم خريطة تدفق Flowchart لإجراء حل مشكلة	- باستخدام عرض تقديمي يتم عرض الأشكال ووظيفة كل شكل وتطبيقه على المثال السابق - باستخدام استراتيجية التعلم التعاوني يتم عرض مثال آخر - كتابة برنامج بسيط لطباعة جملة - "Hello, Kuwait!" على الشاشة. - كتابة الخوارزمية - استخدم الدالة print () لطباعة النص "Hello, Kuwait!" على الشاشة. - رسم خريطة التدفق	- توزيع بطاقات على المجاميع لرسم الخريطة 	10

م	نواتج التعلم	محتوى التعلم (الأنشطة والاستراتيجيات)	التقويم	الزمن
6	- استخدام محرر الأوامر PyCharm	- باستخدام استراتيجية المحاضرة: - يتم التوضيح للطلبة أنه يوجد العديد من مواقع تحرير التعليمات البرمجية التي توفر الأدوات لكتابة التعليمات وتحريرها واختبارها وتصحيح أخطائها في مكان واحد. - وسيتم استخدام محرر PyCharm في منهج الحاسوب لتعلم وتطبيق تعليمات لغة البايثون . - عن طريق جهاز المعلم يتم فتح برنامج PyCharm واستعراضه على Data Show واستعراض أجزاء نافذة المحرر 	- مناقشة المتعلمين حول معلوماتهم السابقة عن مناطق الشاشة : - ما اسم الشريط العلوي في البرنامج وماذا يعرض ؟	5
7	- انشاء مشروع جديد	- باستخدام التطبيق العملي على جهاز المعلم يتم شرح خطوات انشاء مشروع جديد وتحديد اسمه ومكان الحفظ	- بمشاركة المتعلمين يتم إعادة تطبيق العملية على جهاز المعلم	5
8	- انشاء ملف بايثون جديد	- باستخدام استراتيجية العصف الذهني يطلب من المتعلمين استنتاج طريقة انشاء ملف بايثون جديد - ثم يتم شرح الخطوات عمليا على جهاز المعلم مع توضيح مكان تبويب اسم الملف و أرقام أسطر التعليمات البرمجية	- يتم الطلب من أحد المتعلمين انشاء ملف بايثون جديد على جهاز المعلم - أين ظهر تبويب اسم الملف الجديد ؟	5
9	- كتابة برنامج بسيط	- باستخدام السبورة يتم شرح كيفية كتابة تعليمة بسيطة بلغة البايثون - Print (" I'm a Python developer ")	- يتم التطبيق عمليا على أجهزة المتعلمين.	5
1	- تشغيل البرنامج	- من خلال نافذة البرنامج في جهاز المعلم يتم استعراض طرق تشغيل البرنامج (مفاتيح Ctrl+Shift+F10 , الامر Run من القائمة المختصرة لمنطقة كتابة التعليمات البرمجية , الأداة Run من شريط الأدوات)	- يختار المتعلمين احدهم الطرق ويتم تطبيقها لمشاهدة النتيجة على أجهزتهم	5
11	- هياكل البيانات	- من خلال PowerPoint يتم استعراض هياكل البيانات	- يلاحظ المتعلم هياكل البيانات ويميز أقسامها	5
التقويم الختامي		حل ورقة عمل (1-1) ورقة عمل (2-1)		
		20		

**** تذكير المتعلمين بضرورة إغلاق الملف و البرنامج وإعادة تنظيم المختبر .**

اليوم والتاريخ	الفصل	الحصة	التأمل الذاتي		
			الرضا عن تحقق الأهداف	تحديات واجهتني	مقترحات لتحسين الأداء
			☹ ☹ ☹		

التقويم التربوي

06



التقويم التربوي

التقويم التربوي هو عملية شاملة تهدف إلى تقييم وتطوير أداء المتعلمين على مدار فترة زمنية محددة، سواء كانت فصلاً دراسياً أو سنة دراسية كاملة.

هذه العملية ليست فقط لقياس مدى فهم المتعلمين للمواد الدراسية، ولكنها أيضاً أداة لتحسين التعليم من خلال تقديم تغذية راجعة مستمرة وتعديل استراتيجيات التدريس وفقاً للبيانات التي يتم جمعها.

أنواع التقويم حسب الفترة الزمنية

التقويم التشخيصي: يهدف إلى تحديد نقاط القوة والضعف لدى المتعلمين، والفجوات في معرفتهم ومهاراتهم، يتم تطبيقه قبل بداية الوحدة أو الدرس وعليه يمكن المعلمين تعديل خططهم التعليمية لتلبية احتياجات متعلميهم الفردية وضمان تحقيق النجاح.



التقويم التكويني: التقويم المستمر يتم تطبيقه أثناء سير الحصة لتقديم تغذية راجعة مستمرة لكل من المعلمين والمتعلمين بهدف تحسين عملية التعلم والأداء (المناقشة - الملاحظة - التقويم الصفّي - الواجبات).



التقويم الختامي: هو التقويم النهائي للجهد المبذول يتم تطبيقه في نهاية وحدة دراسية أو فصل دراسي أو سنة دراسية يهدف هذا التقويم إلى قياس مدى استيعاب المتعلمين للمهارات التعليمية المحددة وتقييم أدائهم الشامل.





استراتيجيات التقويم التربوي :



1- استراتيجيات التقويم المعتمد على الأداء

التقويم المعتمد على الأداء يركز على تقييم قدرة المتعلمين على تطبيق المعرفة والمهارات في مواقف واقعية. وهو نوع يشجع على التعلم النشط والتفاعل ويعطي صورة دقيقة عن مدى تمكن المتعلمين من المواد الدراسية .

تشمل هذه الاستراتيجية ملاحظة أداء المتعلمين أثناء الأنشطة الصفية والمناقشات.

الملاحظات الصفية
والمناقشات

أ

تتضمن المشاريع تكليف المتعلمين بمهمة تتطلب منهم تطبيق المعرفة والمهارات التي تعلموها.

المشاريع

ب

يتضمن هذا النوع من التقويم إعداد المتعلمين وتقديمهم لعروض حول موضوعات معينة.

العروض التقديمية

ج

تتضمن تقديم مشكلة للمتعلمين لحلها باستخدام مهاراتهم ومعرفتهم.

حل المشكلات:

د



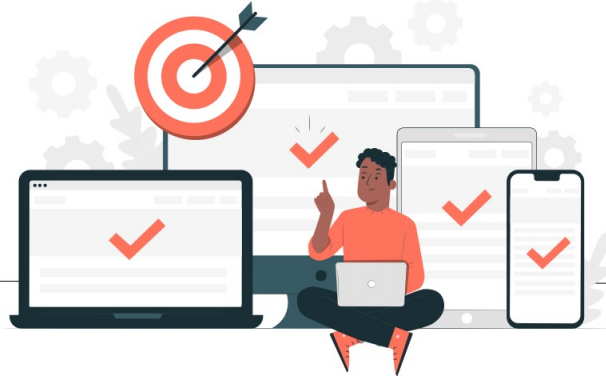
تابع - استراتيجيات التقويم التربوي :



2- استراتيجية التقويم بالتواصل

هو نوع من التقويم يعتمد على الحوار والتفاعل بين المعلم والمتعلمين، أو بين المتعلمين أنفسهم، كوسيلة لتقييم وفهم مدى استيعابهم للمفاهيم والمعارف.

- المناقشات الصفية



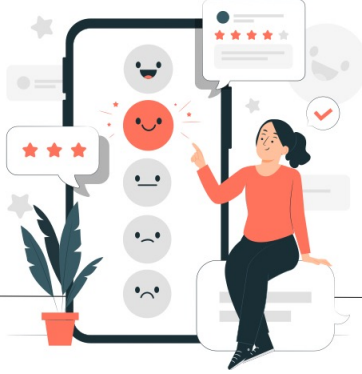
3- التقويم المعتمد على الملاحظة

تعد استراتيجية التقويم المعتمد على الملاحظة من أنواع التقويم النوعي الذي يدون فيه سلوك المتعلمين بهدف التعرف على اهتماماتهم وميولهم واتجاهاتهم وتفاعلاتهم مع بعضهم البعض، بقصد الحصول على معلومات تفيد في الحكم على أدائهم وفي تقويم مهاراتهم وأخلاقياتهم وطريقة التفكير التي ينتهجونها.

- تفعيل سجل المتابعة اليومية



تابع - استراتيجيات التقويم التربوي :



4- استراتيجية مراجعة الذات

تقوم هذه الاستراتيجية على تحويل التعلم السابق إلى تعلم جديد، وذلك بتقييم ما تعلمه المتعلم من خلال تأمله الخبرة السابقة، وتحديد نقاط القوة والنقاط التي بحاجة إلى تحسين، وتحديد ما سيتم تعلمه لاحقاً. وتعد هذه الاستراتيجية مكوناً أساسياً للتعلم الذاتي، و حل المشكلات مما يمكن المتعلم من تشخيص نقاط القوة في أدائه، وتحديد حاجاته وتقييم اتجاهاته.



5- استراتيجية تقويم الأقران

استراتيجية تعليمية يقوم فيها مجموعة من المتعلمين أو أقرانهم الذين يشاركون في العملية التعليمية بتقييم مهام ونشاطات زملائهم. وفقاً لمعايير الجودة التي يضعها المشرف أو المعلم .



توزيع درجات مادة الحاسوب للمرحلة الثانوية (التعليم العام)

للف (العاشر - الحادي عشر علمي- الحادي عشر أدبي – الثاني عشر علمي – الثاني عشر أدبي)

الدرجة الكلية لنهاية العام	الفترة الدراسية الثانية						الفترة الدراسية الأولى					
	الجمالي	الاختبار	المشروع	توزيع درجة الأعمال			الجمالي	الاختبار	المشروع	توزيع درجة الأعمال		
				1	2	3				1	2	3
20		10	4					10	4			
20=2/(20+20)	20	الاختبار العملي	المشروع الحاسوبي	التطبيق العملي لأوراق العمل	أنشطة صفية	الالتزام والتعاون وسلوك المتعلم	20	الاختبار العملي	المشروع الحاسوبي	التطبيق العملي لأوراق العمل	أنشطة صفية	الالتزام والتعاون وسلوك المتعلم

الاختبارات



الاختبارات التحصيلية:

الاختبارات التحصيلية: هي الأداة التي تستخدم في قياس المعرفة والفهم والمهارة لمادة دراسية معينة قياسا كميا بهدف قياس نواتج التعلم لدى المتعلم في شكل اختبار، بحيث يراعي فيه قواعد القياس والتقنين من ثبات وصدق.

أغراض الاختبارات التحصيلية

1. التعرف على ما يمتلكه المتعلم من المهارات اللازمة لتعلم محتوى دراسي جديد.
2. الوقوف على مواطن القوة والضعف لدى المتعلم.
3. تصنيف المتعلمين في مجموعات وقياس مستوى تقدمهم بالمادة.
4. تقدير المستوى العام لتحصيل المتعلمين في مادة دراسية معينة.
5. تنشيط دافعية التعلم والنقل من صف لآخر ومنح الدرجات والشهادات.

كيفية التخطيط لبناء الاختبارات التحصيلية

إعداد نموذج الإجابة وتوزيع الدرجات		تحديد موضوع الاختبار	
وضع تعليمات الاختبار		تحديد أهداف الاختبار	
مراجعة وتدقيق الاختبار		تحديد الأوزان النسبية لكل هدف. (جدول المواصفات)	
إجراء وتنفيذ الاختبار		تحديد شكل الاختبار	
تصحيح الاختبار		صياغة فقرات الاختبار	
تحليل نتيجة الاختبار		ترتيب الأسئلة	
تقويم الاختبار			



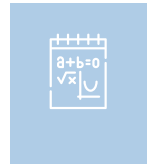
الميزان النسبي / جدول المواصفات للاختبار

هو جدول يوضع لبيان الوزن النسبي لكل موضوع بالنسبة للمقرر ومن ثم توزيع درجات الاختبار على المواضيع مقارنة بأوزانها النسبية.

وضع جدول المواصفات لضمان أن يكون الاختبار شاملاً لكل المواضيع ومتوازناً. فيعد بالشكل التالي:

- تحديد موضوعات المادة التي يراد قياس تحصيل المتعلم فيها.
- تحديد عدد الحصص اللازمة لتدريس كل موضوع.
- تحديد الوزن النسبي لموضوعات المادة الدراسية ويمكن الاستفادة بإعداده من خلال المعادلة التالية:

$$\text{الوزن النسبي لأهمية الموضوع} = \frac{\text{عدد الحصص اللازمة لتدريس الموضوع} \times 100}{\text{عدد الحصص اللازمة لتدريس المادة}}$$



خطوات إعداد الاختبار

- يتم تكليف كل معلم رسمياً بعمل اختبار معين وتحدد له فترة زمنية لتسليمه لرئيس القسم مع نموذج الإجابة حسب الميزان النسبي المتفق عليه.
- يتسلم رئيس القسم الاختبار ويراجعه ويغير ما هو ضروري.
- يعرض رئيس القسم الاختبار على الموجه الفني للاعتماد.
- يوضع الاختبار في ظرف مغلق ومختوم في مكان أمين بالمدرسة بمعرفة رئيس القسم.
- تسحب الأسئلة وتدبس وتوضع في أظرف مغلقة ومختومة في مكان أمين بالمدرسة تحت إشراف رئيس القسم.
- تحفظ الأسئلة.
- تتلف الأوراق الزائدة.



ملاحظات الواجب إتباعها عند وضع الأسئلة

- ينبغي دراسة محتوى المنهج وأهدافه دراسة وافية حتى تأتي الأسئلة محققة للأهداف التي وضع المنهج وأهدافه الواجب تحقيقها.
- يراعى في الأسئلة أن تكون شاملة لموضوعات المنهج وموزعة على أجزائه المختلفة.
- يراعى التنوع في ما تقيسه الأسئلة (من مهارات) الحفظ، التذكر، التحليل، الاستنتاج.
- يراعى أن تكون الأسئلة السهلة والصعبة قليلة، أما أغلب الأسئلة متوسطة.
- ينبغي أن تكون الأسئلة واضحة العبارة خالية من اللبس والغموض.
- ينبغي أن تكون الأسئلة لها إجابة واحدة.
- توضع درجة كل بند في مربع واضح أمامه.

تصحيح الاختبار

هناك إرشادات تساعد المعلم على القيام بهذه المهمة على أسس علمية وبطريقة سريعة منها:

1. وجود نموذج للإجابات لأسئلة الاختبار، محدد فيها توزيع الدرجات على جزئيات السؤال.
2. مراعاة الدقة أثناء التصحيح.
3. الالتزام بنظام معين لتقدير الدرجات ليكون منصفاً في تقييم عمل المتعلمين.
4. التغلب على العامل الشخصي عند تقدير الدرجات وعدم توقف علامة المُتعلّم على من يصحح ورقته.
5. التركيز على إجابة المُتعلّم وليس على الخلفية السابقة لمستواه العلمي.
6. ضرورة توثيق ملفات اختبارات المعلمين بأخذ نسخ منها على وسيط تخزيني خارجي، ثم توضع نسخه منه على قرص في سجل الاختبارات بالقسم.
7. المراجعة العلمية والفنية لجميع أوراق المتعلمين بعد انتهاء التصحيح، ويقصد بالمراجعة إعادة تصحيح وليس فقط التأكد من جمع الدرجات.



الميزان النسبي المقترح: لدرجات الامتحان العملي للفترة الدراسية الأولى

لمادة تقنية المعلومات - الصف العاشر

م	الأسبوع	المهارات	الدرجة
1	الأسبوع الأول الأمن السيبراني	• نظري	-
2	الأسبوع الثاني برمجة python مدخل إلى البرمجة	• فتح البرنامج • فتح / حفظ الملف • انشاء ملف بايثون جديد • اعداد خطة الحل الخوارزمية Algorithm • انشاء خريطة التدفق Flowchart	1
3	الأسبوع الثالث المتغيرات Variables	• الإعلان عن متغير • دالة الإخراج print() • العمليات الحسابية • دالة input()	2
5	الأسبوع الرابع + الخامس السلاسل النصية Strings	• عمليات المقارنة • العمليات الشرطية if, elif, else	2
6	الأسبوع السادس الشروط Conditions	• الدالة while • الدالة for • الدالة Range	1
7	الأسبوع السابع + الثامن التكرار Loops	• انشاء سلسلة نصية • الفهرسة Indexing[] • القطع Slicing • جمع السلاسل النصية • تكرار عناصر السلسلة النصية • دوال السلاسل النصية: • طول السلسلة النصية len() • capitalize() • upper() • lower() • find() • replace() • تعليمية f-string • int() • float() • center()	2



الميزان النسبي المقترح: لدرجات الامتحان العملي للفترة الدراسية الأولى

لمادة تقنية المعلومات - الصف العاشر

الدرجة	المهارات	الأسبوع	م
2	مفهوم الأخطاء (Errors) والاستثناءات (Exceptions) • أنواع الأخطاء الشائعة Errors . - أخطاء نحوية (Syntax Errors). - أخطاء وقت التشغيل (Runtime Errors). - أخطاء المنطق (Logic Errors). تابع / تصحيح الأخطاء و الاستثناءات • التعامل مع الاستثناءات Exceptions. • التعليم البرمجية try / except • تصحيح الأخطاء Debugging	الأسبوع التاسع + العاشر تصحيح الأخطاء والاستثناءات Debugging and Exceptions	8
الإجمالي = 10			

الخاتمة

نختتم هذا الدليل بتفاؤل وامل في ان يكون له الأثر الإيجابي على رحلتكم التعليمية في تدريس منهج الصف العاشر وأن يكون معينا للمعلم أثناء تقديم المحتوى العلمي , فقد تم تصميم هذا الدليل لمساعدة المعلم في تقديم محتوى الامن السيبراني من خلال فيديوهات معدة وجاهزة لاستخدام المعلم يمكن أن يستعين بها لتقديم المحتوى العلمي وعرضه للمتعلمين كوسيلة مساعدة أثناء الشرح , وهي غير ملزمة له , ويترك للمعلم حرية الاختيار في اعداد وتصميم ما يناسبه من فيديوهات .

وندعو جميع المعلمين للاستفادة القصوى من الموارد والمعلومات المقدمة في هذا الدليل بهدف اثراء مهاراتهم وتعزيز معلوماتهم وخبراتهم , ونتطلع بشغف الى رؤية تفاعلاتهم ومساهماتهم في هذا المجال .

نحن ملتزمون بدعمكم في رحلتكم التعليمية التي ستستمر بفضل جهودكم المخلصة في تقديم الأفضل لمتعلميكم وحرصكم على استكشاف وتطوير مهاراتهم والاطلاع على كل ما هو جديد في مجال التكنولوجيا والبرمجة , حيث تحتم علينا ظروف العالم المتسارعة والتطور السريع في مجال مادة تقنية المعلومات الى ضرورة مواكبة كل ما هو جديد على الساحة .

نتطلع الى رؤية نتائجكم المذهلة في مجال البرمجة متمثلة في نتائج المتعلمين من مشاريع جيدة تعكس مدى استفادة المتعلمين من معلومات أثناء الفصل الدراسي , وندعوكم الى المشاركة النشطة في المسابقات والتحديات التقنية لتطوير المهارات البرمجية لكل من المعلم والمتعلم .

نتمنى لكم كل التوفيق في رحلتكم التعليمية والمهنية ونأمل أن يكون هذا الدليل بمثابة مورد ومرجع ملهم لكم عند الحاجة.