جَامعة المَنارة
MANARA UNIVERSITY

كلية الهندسة المعلوماتية

برمجة 3
Java Programming

ا. د. علي عمران سليمان

محاضرات الأسبوع الأول
Introduction
الفصل الآول 2024-2025

Contents 1

1-Java Primer.

1.1 Comments, Base Types

1.2 Classes and Objects (Defining, Access Control Modifiers, static, abstract, final Modifier), Declaring , (Instance Variables, Methods, Parameters, Constructors, main Method)

1.3 Strings, Wrappers, Arrays, and Enum Types (Concatenation, StringBuilder Class, Wrapper Types, Automatic Boxing Unboxing).

1.4 Arrays, Enum.

1.5 Expressions.

1.6 Simple Input and Output (Simple Output Methods, Scanner Class).

1.7 Packages and Imports.

1.8 Software Development.

References

- Deitel & Deitel, Java How to Program, Pearson; 10th Ed(2015)
- د.علي سليمان، بنى معطيات بلغة JAVA، جامعة تشرين 2013-2014

Contents 2

2- Object-Oriented Design

2.1 Goals, Principles and Patterns
2.2 Inheritance
2.3 Interfaces and Abstract Classes
2.4 Exceptions
2.5 Casting and Generics
2.6 Nested Classes
2.7 Exercises

3- GUI

3.1 Introduction
3.2 Overview of Swing Components
3.3 JLabel
3.4 Event Handling
3.5 TextFields
3.6 How Event Handling Works

References

- Deitel & Deitel, Java How to Program, Pearson; 10th Ed(2015)
- د.علي سليمان، بنى معطيات بلغة JAVA، جامعة تشرين 2013-2014

Contents 3

3.7 JButton
3.8 JCheckBox and JRadioButton
3.9 JComboBox
3.10 JList
3.11 Multiple-Selection Lists
3.12 Mouse Event Handling
3.13 Adapter Classes
3.14 Key Event Handling
3.15 Layout Managers (FlowLayout,BorderLayout,GridLayout)
3.16 Panels

References

- Deitel & Deitel, Java How to Program, Pearson; 10th Ed(2015)
- د.علي سليمان، بنى معطيات بلغة JAVA، جامعة تشرين 2013-2014

What IS Object Oriented Programming ?

- البرمجة غرضية التوجه (Object Oriented Programming OOP) هي نموذج برمجة programming paradigm يعتمد على مفهوم "الكائنات".
- نموذج البرمجة: هو نمط style من البرمجة او طريقة way للتفكير في بناء البرمجيات.
- لا يرتبط نموذج البرمجة بلغة معينة بل هو طريقة لبناء برنامج أو منهجية methodology للتطبيق.
- تسهل بعض اللغات الكتابة في بعض النماذج دون غيرها.
- تسمح بعض لغات البرمجة للمبرمج بتطبيق أكثر من نموذج واحد C++.

Programming Paradigms

- The programming paradigm refers to a way of conceptualizing and structuring the tasks a computer performs.

Paradigm	Languages	Description
Procedural	BASIC, Pascal, COBOL, FORTRAN, Ada	Emphasizes linear steps that provide the computer with instructions on how to solve a problem or carry out a task
Object-oriented	Smalltalk, C++, Java	Formulates programs as a series of objects and methods that interact to perform a specific task
Declarative	Prolog	Focuses on the use of facts and rules to describe a problem
Functional	LISP, Scheme, Haskell	Emphasizes the evaluation of expressions, called functions
Event-driven	Visual Basic, C#	Focuses on selecting user interface elements and defining event-handling routines that are triggered by various mouse or keyboard activities

Some notes about Java 1

- يتضمن أي ملف مكتوب بلغة Java صنفاً class عاماً فقط(أو واجهه)، وقد يتضمن البرنامج عددًا كبيرًا من الأصناف، والملف الواحد يملك صنف عام يخزن الملف باسمه على القرص.
- قد يحتوي أي برنامج على عدد كبير من المناهج methods، واحدة منها هي المنهج الرئيس main method، وإذا اشتملت على منهج واحدة فقط، فسيكون هو المنهج الرئيس ، والتي يبدأ منه التنفيذ ويعرف بنقطة الدخول إلى البرنامج.
- يتضمن الصنف (والذي يشبه المخطط blueprint) وصفًا للبيانات الاعضاء data member، حقول البيانات, data fields،خصائص البيانات properties of data والمعروف باسم السمات attribute،والمناهج الاعضاء member method، أو العمليات operations أو الأحداث Actions، والمعروفة بالسلوكيات behaviors (وتؤمن التواصل مع بقية الأصناف).
- يتم اشتقاق الكائنات Objects من الاصناف وتُعرف بالأمثال instances.
- يفضل تضمين التعليقات comments التي تسهل تذكرومعرفة عمل البرنامج.

- عند التصريح عن المناهج والبيانات ، تبدأ عادتاً بمحددات الوصول access specifiers.
- يكتب كود البرنامج داخل جسم الصنف.
- الأصناف هي مصدر الكائنات.
- تتم كتابة برامج Java بلغة عالية المستوى بامتداد java. ويتم مطابقتها لتحويلها إلى لغة وسيطة تُعرف باسم bytecodes وتخزينها بامتداد class.ومن ثم تشغيلها للحصول على النتائج.

- في معظم لغات البرمجة، قد تظهر الأخطاء أثناء:
  - مطابقة البرامج وتخطي قواعد اللغة، والمعروف باسم syntax error.
  - عند تشغيل البرنامج، وإدخال قيم خطأ مثلاً، والمعروف باسم runtime error.
  - إعطاء نتائج غير متوقعة، خطأ معالجة يُعرف بالخطأ المنطقي logical error.

An overview of the Java programming language

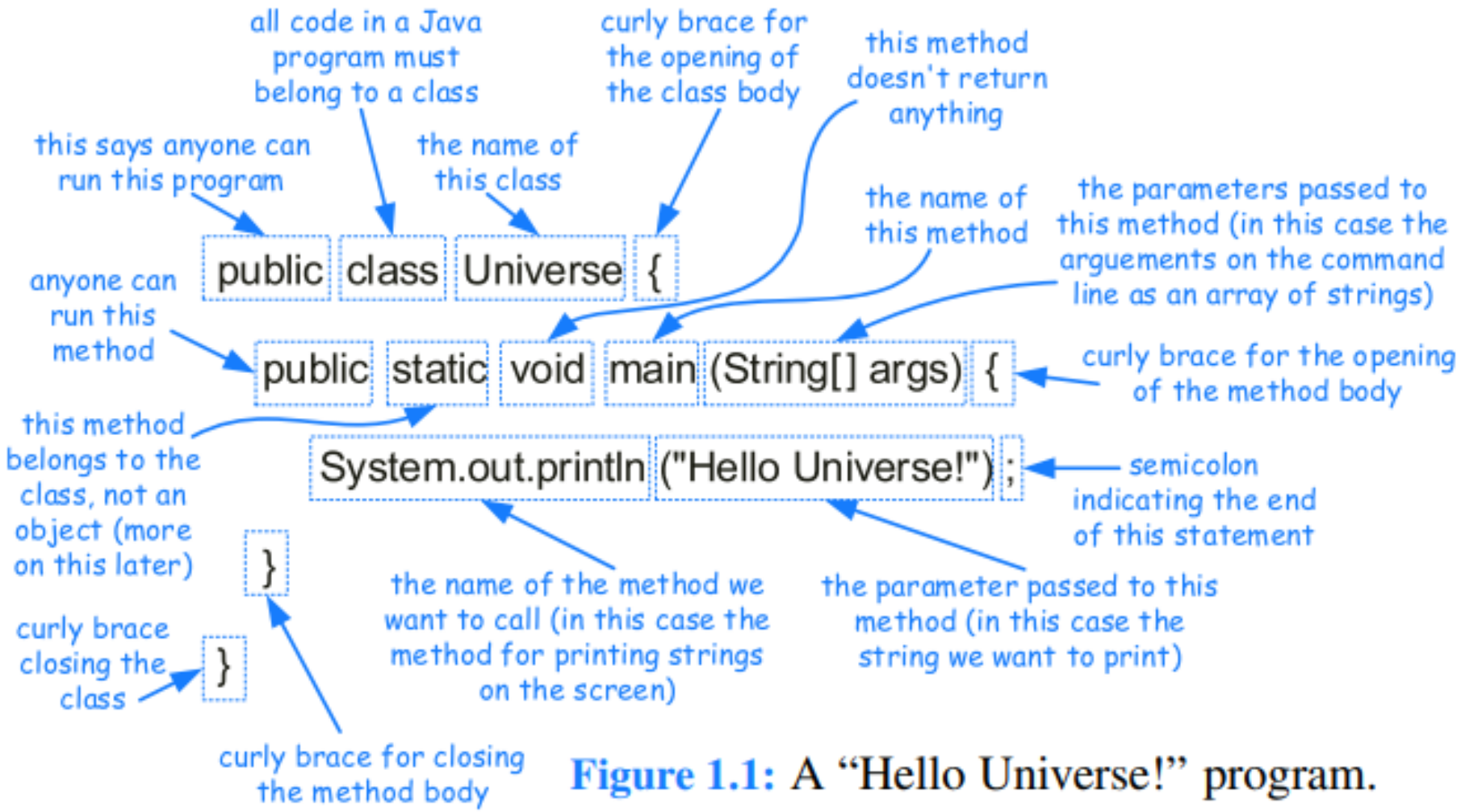


Figure 1.1: A "Hello Universe!" program.

- لإنشاء كائن من صنف يتم استدعاء المنهج الباني من خلال المعامل new في الجافا يكون الحجز ديناميكياً فقط:

```
Scanner input = new Scanner (System.in);
```

  - نجد ثلاثة أحداث:

1. سيتم البحث ضمن الذاكرة عن أماكن متماسة كافية للغرض وحجزها (يتم تخصيص مثيل جديد ديناميكيًا في الذاكرة).
2. يتم تهيئة جميع متغيرات الحالة بالقيم الافتراضية القياسية. null للمتغيرات المرجعية reference و0 لكل الانواع الأساسية، باستثناء المتغيرات المنطقية Boolean (التي تكون افتراضيًا false). وقد يقوم الباني بتعيين قيم أكثر أهمية (غير الافتراضية) لأي من متغيرات الحالة، وإجراء أي عمليات حسابية إضافية يجب إجراؤها عند إنشاء هذا الكائن.
3. بعد تنفيذ الباني يُرجع **returns** المُعامل new مرجعًا reference (أي عنوان بداية الحجز للكائن في الذاكرة) إلى الكائن الذي تم إنشاؤه حديثًا في مثالنا input.

  - وإذا لم يجد حجماً كافياً فسيتم إعادة **null** تخزين هذا العنوان في متغير الكائن.

access control modifiers

- التحكم بمُعدِّلات الوصول المختلفة (من يستطيع الوصول إليها):
- public **class** محدد وصول عام للصنف: أي أن جميع الأصناف يمكنها الوصول إلى الصنف العام، يجب تحديد كل صنف عامة في ملف منفصل باسمه، مثلاً **classname.java**.
- protected **class** محدد وصول محمي للصنف: أي أن الوصول إليه يُمنح فقط للمجموعات التالية من الأصناف الأخرى:
  - الأصناف االوارثة لهذا الصنف المحمي.
  - الأصناف التي تنتمي إلى نفس الحزمة التي ينتمي لها الصنف المحمي.
- private **class** الصنف الخاص: أن الوصول إلى أعضائه يتم فقط للتعليمات البرمجية الموجودة داخل تلك الصنف. لا يمكن للأصناف الفرعية (الوارثة) ولا أي أصناف أخرى الوصول إلى أعضاء هذا الصنف.
- في حالة عدم وجود أي معدِّل صريح للتحكم بالوصول، فستكون الحالة الافتراضية default (friendly)له وتُعرف بمستوى الوصول الخاص بالحزمة package-private. يسمح للأصناف في نفس الحزمة بالوصول إلى أعضائه.

The static Modifier

- static modifier في Java يمكن أن يحدد لأي كائن Objectأو منهج Method ضمن صنف .
- عندما يتم الإعلان عن كائن أومتغير من صنف على أنه static، فإن قيمته ترتبط بالصنف ككل، بدلاً من ارتباطها مع كل مثيل بمفرده مشتق من الصنف، أي تُستخدم المتغيرات staticلتخزين معلومات global عن الصنف.
- عندما يتم الإعلان عن منهج من صنف ما على أنه static، فإنه يرتبط أيضًا بالصنف نفسه، وليس بكائن معين من الصنف. عادةً ما يتم استدعاؤه باستخدام اسم الصنف نقطه اسم المنهج.
- abstract يمكن الإعلان عن منهج من صنف ما على أنه مجرد، وفي هذه الحالة يتم توفير توقيعها أو بصمتها signature ولكن بدون تنفيذ implementation جسم المنهج.
- final يمكن تهيئة المتغير الذي تم التصريح عنه بالمعدِّل final كجزء من الإعلان عنه، ولكن لا يمكن أبدًا تعيين قيمة جديدة له تعتبر قيمتة ثابته وهو ضمنا static، وإذا كان متغير مرجعي reference، فسيشير دائمًا إلى نفس مكان الكائن ولكن يمكن تغيير محتوى الكائن.
- Final للمناهج لا يمكن أن يعمل لها overridden سيتم معالجتها في الوراثة .

Data Hiding

- يخفي الكائن حقوله الداخلية الخاصة عن التعليمات البرمجية الموجودة خارج الصنف وبالتالي لايمكن استخدامها.

- فقط مناهج الصنف يمكنها الوصول مباشرة إلى البيانات الداخلية للكائن وتغييرها، ولكي نصل إليها من خارج الصنف يجب أن تكون عامة، وبالتالي يمكن وضع الاحتياطات المناسبة عند برمجتها لمنع المستثمر من عمل غير منطقي أي حماية المعطيات من الاستثمار الخاطئ.

- Data hiding إخفاء البيانات: يعد أمرًا مهمًا لأن الأصناف تُستخدم عادةً كمكونات في أنظمة البرامج الكبيرة، والتي تضم فريقًا من المبرمجين ويستخدمها العديد من المستثمرين.

- يساعد إخفاء البيانات في تعزيز integrity صحة وتكامل البيانات الداخلية للكائن.

declaring instance variables

- **general syntax الشكل العام للتصريح عن متغير مثل أو أكثر من صنف يكون بالشكل التالي:**

[*modifiers*] *type identifier1*[=*initialValue1*], *identifier2*[=*initialValue2*];

مثال:

private int count;

- **حيث** private **هي نمط الوصول** modifier.
- int **هي نوع المتغير.**
- count **اسم المتغير.**

- **نظراً لعدم التعريف والتجهيز بقيمة، بشكل تلقائي سيتم اسناد القيم الافتراضية صفر لإنه عدد كما مر سابقاً.**

declaring method

- general syntax الشكل العام للتصريح عن منهج من صنف يكون بالشكل التالي:

[*modifiers*] *returnType methodName(type1 param1 ,..., typen paramn*)
{ // method body . . . }

مثال:

public void increment(int delta) { count += delta; }

- modifiers **such as** public, private, **and** protected.
- returnType **defines the type of** value returned **by the method**.
- methodName **can be any valid Java identifier**.
- The list of parameters (or **Parameter variable declaration**) **and their** types declares **the local variables that are to be passed as arguments to this method.**

declaring constructor

- constructor الباني هو نوع خاص special type من مناهج الصنف وموجود افتراضيًا , تقوم java ومعظم اللغات ببنائة.
- يتم استدعاء الباني المكتوب تلقائيًا called automatically عند إنشاء كائن وإذا لم يكتب ينادى الافتراضي.
- يتم استخدامه لتهيئة حقول الكائن الذي يتم إنشاؤه من الصنف بحيث يكون في حالة أولية منسجمة ومستقرة بالقيم الإقتراضية الأولية. أو بأية قيم مرغوبه تمرر له.
- يمكنه إجراء عمليات حسابية أكثر تعقيدًا من إسناد القيم في وقت إنشاء الكائن من خلال تضمينها في الباني.
- الصيغة العامة للتصريح عن مُنشئ في Javaهي كما يلي:

modifiers name(*type0 parameter0* , ..., *typen*−1 *parametern*−1)

{ // constructor body . . . }

- modifiers **typically public And can be** protected, private**, or the default** package-level **visibility** But **cannot be** static, abstract, **or** final.

- **We don't specify a** return type **for a constructor** (**not even void**).

-**The name must be the same name as the class**.

Counter d = new Counter(5);

The Keyword this

- هناك ثلاثة أسباب شائعة لضرورة استخدام this (المرجع) من داخل جسم الطريقة:

1. لتخزين المرجع في متغير، أو إرساله كمعامل إلى منهج أخرى يتوقع مثيلاً (مرجعاً) من هذا النوع كوسيطة. (نقول أن المتغير المحلي يخفي متغير الحالة).

2. للتمييز بين متغير حالة ومتغير محلي بنفس الاسم.

3. للسماح لجسم باني واحد باستدعاء جسم باني آخر.

Objects

- **Object-oriented programming** (**OOP**) is a programming paradigm based on the concept of "objects"

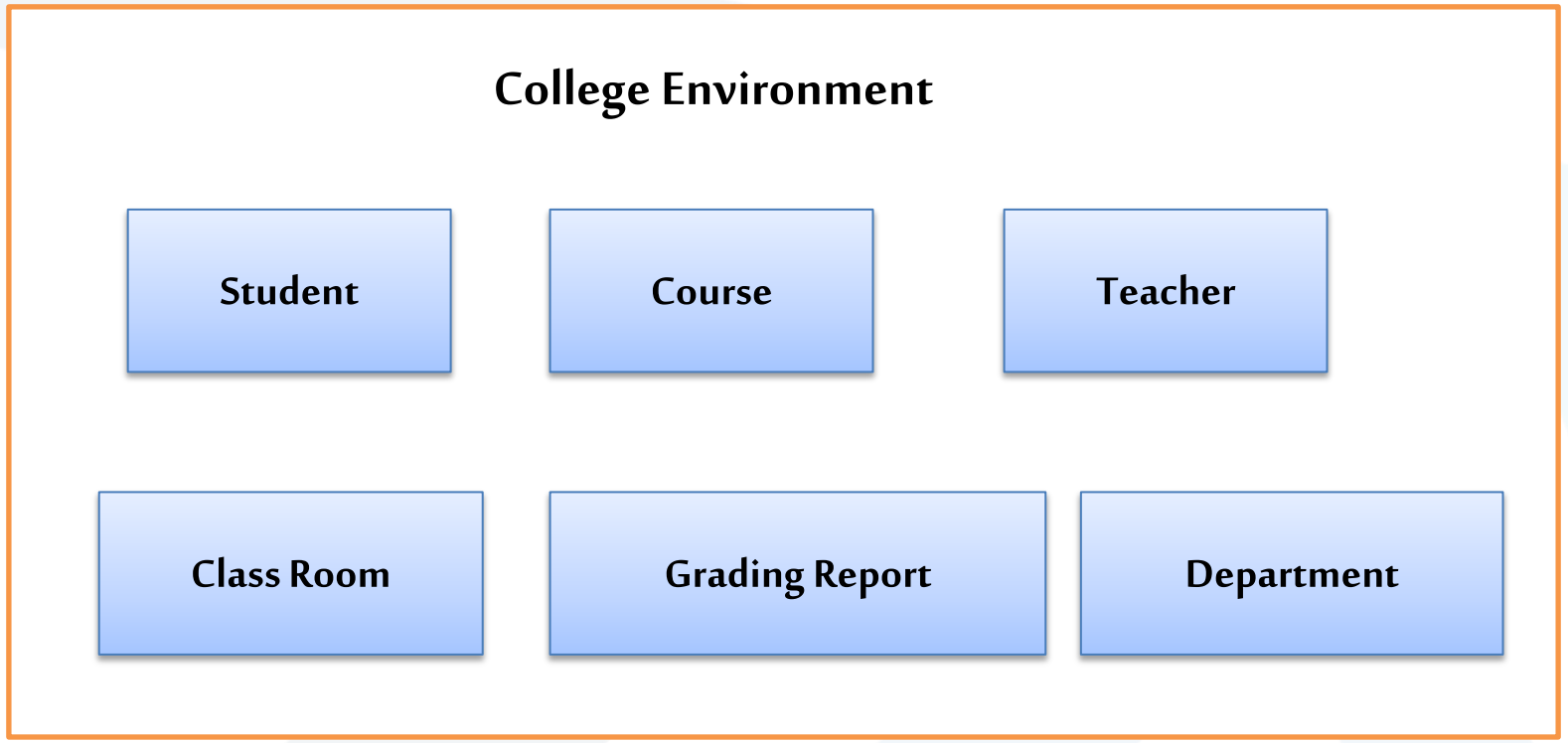
Object : is a thing (Tangible – Intangible)



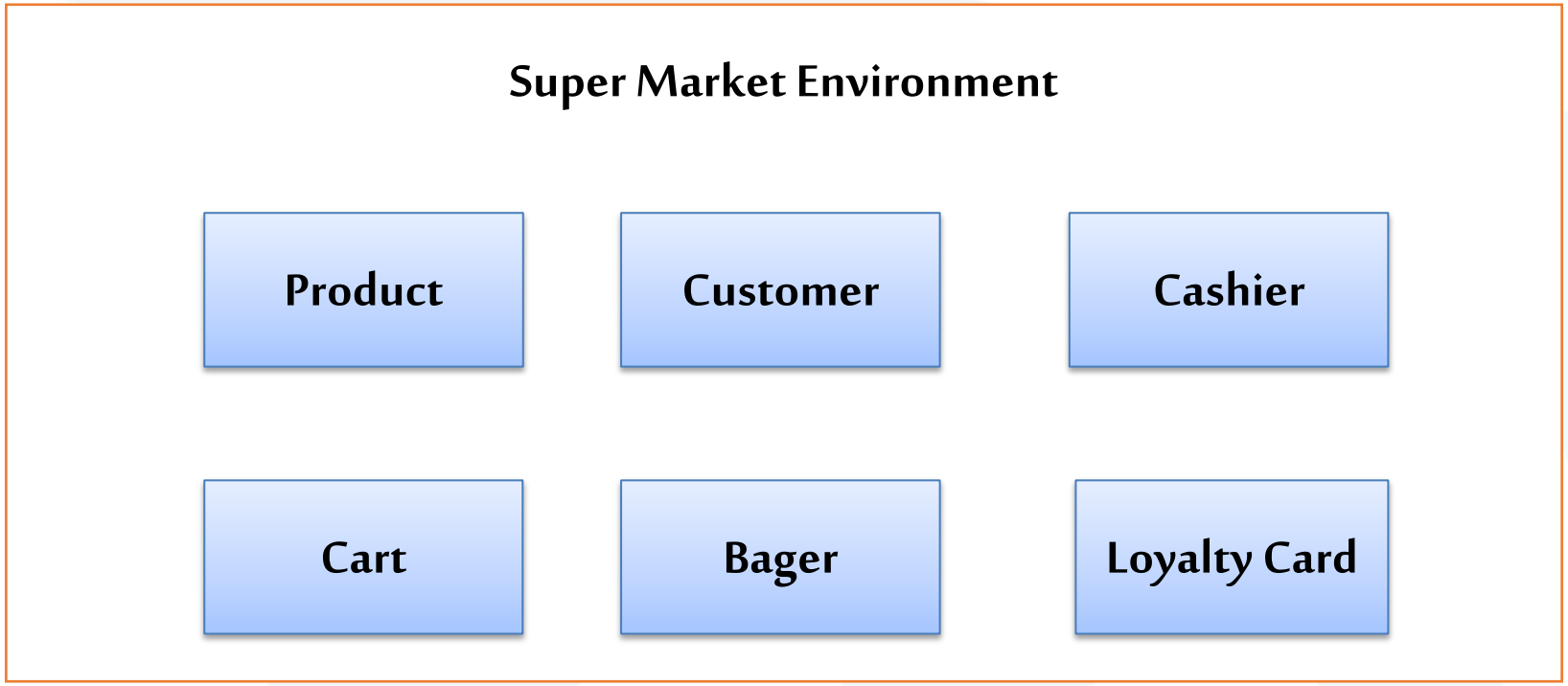




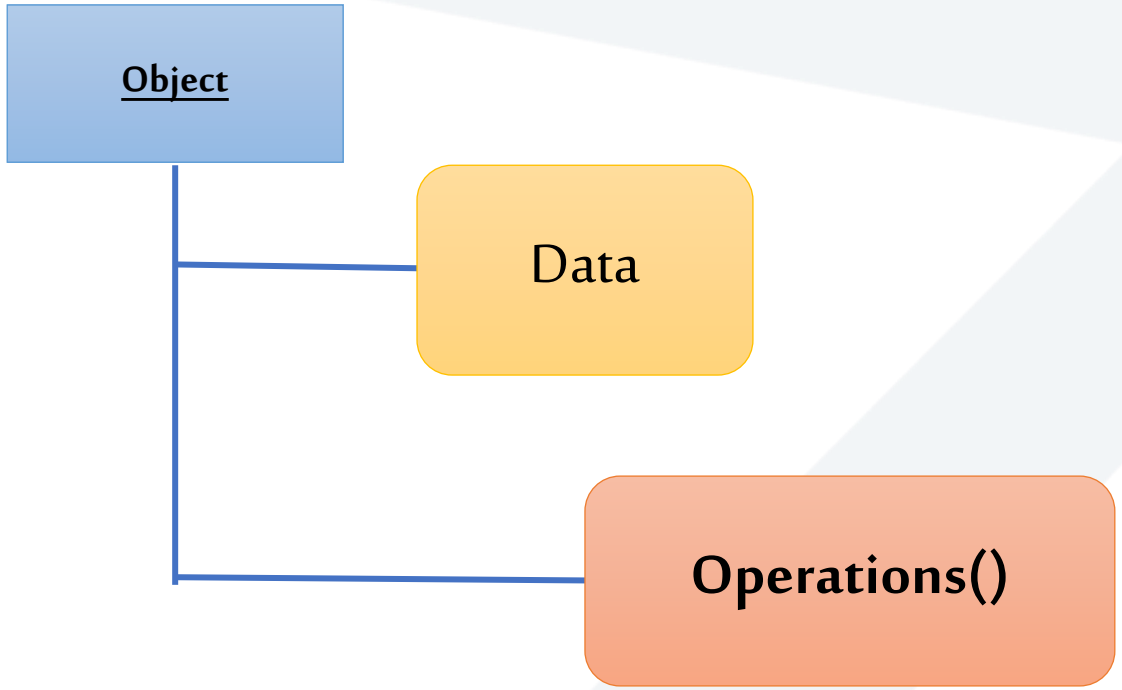
Objects in College Management Program



Objects in Super market Program



Object Is comprised Of ?



Product Object

Data

1- Product _name
2- Product _code
3- Price
4- Producer
5-Discount

Operations ( )

1- ModifyPrice ()
2- SetDiscount ()
3- GetProductName ()
4- GetProductPrice ()

Object Is comprised Of ?

StudentObject

Data

1- Student_name

2- University_Id

3- Birth_Date

4- Address

5-GPA

6- Study_Level

Operations ( )

1- Modify GPA()

2- Change Study level ()

3- Get Student Name ()

4- Get Student Address ()

Car Object

Data

1- Factory

2- Model

3- Fuel_Capacity

4- No_of_doors

5-Color

6- Shape

Operations ( )

1- Set Factory Name()

2- Change Color ()

3- Get Car Info ()

4- .....

What is Class ? Why we need It ?

Student 1	Student 2	Student 3
Data:	**Data:**	**Data:**
1- Student_name	1- Student_name	1- Student_name
2- University_Id	2- University_Id	2- University_Id
3- Birth_Date	3- Birth_Date	5-GPA
4- Address	4- Address	6- Study_Level
5-GPA	6- Study_Level	
6- Study_Level		
Operations ( )	**Operations ( )**	**Operations ( )**
1- Get Student Name ()	1- Get Student Name ()	1- Get Student Name ()
2- Change Study level ()	2- Change Study level ()	2- Change Study level ()
3-Modify GPA()	4- Get Student Address ()	3- Modify GPA()
4- Get Student Address ()		5- Get University_Id ()

What is Class ? Why we need It ?

Class Student	Student 1	Student 2	Student 3
Data: 1- Student_name 2- University_Id 3- Birth_Date 4- Address 5-GPA 6- Study_Level	**Data:** 1- Student_name 2- University_Id 3- Birth_Date 4- Address 5-GPA 6- Study_Level	**Data:** 1- Student_name 2- University_Id 3- Birth_Date 4- Address 6- Study_Level	**Data:** 1- Student_name 2- University_Id 5-GPA 6- Study_Level
Operations ( ) 1- Get Student Name () 2- Change Study level () 3-Modify GPA() 4- Get Student Address ()	**Operations ( )** 1- Get Student Name () 2- Change Study level () 3-Modify GPA() 4- Get Student Address ()	**Operations ( )** 1- Get Student Name () 2- Change Study level () 4- Get Student Address ()	**Operations ( )** 1- Get Student Name () 2- Change Study level () 3- Modify GPA() 5- Get University_Id ()

What is Class ? Why we need It ?

Class Student	Student 1	Student 2	Student 3
Data:	**Data:**	**Data:**	**Data:**
1- Student_name	1- Student_name	1- Student_name	1- Student_name
2- University_Id	2- University_Id	2- University_Id	2- University_Id
3- Birth_Date	3- Birth_Date	3- Birth_Date	5-GPA
4- Address	4- Address	4- Address	6- Study_Level
5-GPA	5-GPA	6- Study_Level	
6- Study_Level	6- Study_Level		
7- Email	7- Email	7- Email	7- Email
Operations ( )	**Operations ( )**	**Operations ( )**	**Operations ( )**
1- Get Student Name ()	1- Get Student Name ()	1- Get Student Name ()	1- Get Student Name ()
2- Change Study level ()	2- Change Study level ()	2- Change Study level ()	2- Change Study level ()
3-Modify GPA()	3-Modify GPA()	4- Get Student Address ()	3- Modify GPA()
4- Get Student Address ()	4- Get Student Address ()		5- Get University_Id ()

What is Class ? Why we need It ?

Class Student	Student 1	Student 2	Student 3
Data:	**Data:**	**Data:**	**Data:**
1- Student_name	1- Student_name	1- Student_name	1- Student_name
2- University_Id	2- University_Id	2- University_Id	2- University_Id
3- Birth_Date	3- Birth_Date	3- Birth_Date	5-GPA
4- Address	4- Address	4- Address	6- Study_Level
5-GPA	5-GPA	6- Study_Level	
6- Study_Level	6- Study_Level		
7- Email	7- Email	7- Email	7- Email
Operations ( )	**Operations ( )**	**Operations ( )**	**Operations ( )**
1- Get Student Name ()	1- Get Student Name ()	1- Get Student Name ()	1- Get Student Name ()
2- Change Study level ()	2- Change Study level ()	2- Change Study level ()	2- Change Study level ()
3-Modify GPA()	3-Modify GPA()	4- Get Student Address ()	3- Modify GPA()
4- Get Student Address ()	4- Get Student Address ()		5- Get University_Id ()
5- Print Student Info ()	5- Print Student Info ()	5- Print Student Info ()	5- Print Student Info ()

What is Class ? Why we need It ?

Class Student

Data:

1- Student_name
2- University_Id
3- Birth_Date
4- Address
5-GPA
6- Study_Level
7- Email

Operations ( )

1- Get Student Name ()
2- Change Study level ()
3-Modify GPA()
4- Get Student Address ()
5- Print Student Info ()

What is Class ? Why we need It ?

Class Student

Data:

1- Student_name
2- University_Id
3- Birth_Date
4- Address
5-GPA
6- Study_Level
7- Email

Operations ( )

1- Get Student Name ()
2- Change Study level ()
3-Modify GPA()
4- Get Student Address ()
5- Print Student Info ()

Student 1

Data:

1- Student_name
2- University_Id
3- Birth_Date
4- Address
5-GPA
6- Study_Level
7- Email

Operations ( )

1- Get Student Name ()
2- Change Study level ()
3-Modify GPA()
4- Get Student Address ()
5- Print Student Info ()

What is Class ? Why we need It ?

Class Student

Data:

1- Student_name
2- University_Id
3- Birth_Date
4- Address
5-GPA
6- Study_Level
7- Email

Operations ( )

1- Get Student Name ()
2- Change Study level ()
3-Modify GPA()
4- Get Student Address ()
5- Print Student Info ()

Student 1

Data:

1- Student_name =Adam
2- University_Id =1232024
3- Birth_Date =20/10/2003
4- Address =Hama TTT
5-GPA = 3.5
6- Study_Level = 6
7- Email = Adam@YY.com

Operations ( )

1- Get Student Name ()
2- Change Study level ()
3-Modify GPA()
4- Get Student Address ()
5- Print Student Info ()

What is Class ? Why we need It ?

Class Student

Data:

1- Student_name
2- University_Id
3- Birth_Date
4- Address
5-GPA
6- Study_Level
7- Email

Operations ( )

1- Get Student Name ()
2- Change Study level ()
3-Modify GPA()
4- Get Student Address ()
5- Print Student Info ()

Student 1

Data:

1- Student_name =Adam
2- University_Id =1232024
3- Birth_Date =20/10/2003
4- Address =Hama TTT
5-GPA = 3.5
6- Study_Level = 6
7- Email = Adam@YY.com

Operations ( )

1- Get Student Name ()
2- Change Study level ()
3-Modify GPA()
4- Get Student Address ()
5- Print Student Info ()

Student 1

Data:

1- Student_name =Sam
2- University_Id =1242025
3- Birth_Date =10/5/2003
4- Address =homs III
5-GPA = 3.6
6- Study_Level = 6
7- Email = Sam@YY.com

Operations ( )

1- Get Student Name ()
2- Change Study level ()
3-Modify GPA()
4- Get Student Address ()
5- Print Student Info ()

انتهت محاضرة الأسبوع الأولى