CSC 1252 · PRINCIPLES OF PROGRAMMING 1

Chapter 2

Elementary Programming

Summary + Quiz with answers

مبادئ البرمجة 1

الفصل 2

أساسيات البرمجة

ملخّص + اختبار مع الإجابات

This chapter teaches the building blocks of every program: reading input, identifiers, variables, constants, numeric data types, operators and expressions, type conversion, and the steps of the software development process.

يقدّم هذا الفصل اللبنات الأساسية لأي برنامج: قراءة المدخلات، والمعرّفات، والمتغيّرات، والثوابت، وأنواع البيانات العددية، والعوامل والتعابير، وتحويل الأنواع، ومراحل عملية تطوير البرمجيات.

Part 1 — Summary

الجزء الأول — الملخّص

1. Reading input from the console

1. قراءة المدخلات من لوحة المفاتيح

- Create a Scanner: `Scanner input = new Scanner(System.in);`
- Read values: `nextInt()`, `nextDouble()`, `nextLong()`, `next()` (a word), `nextLine()` (a whole line).
- **Explicit import**: `import java.util.Scanner;` — **implicit (wildcard) import**: `import java.util.*;`
- Always print a **prompt** before reading so the user knows what to type.

- أنشئ كائن Scanner : `Scanner input = new Scanner(System.in);`
- اقرأ القيم بـ: `nextInt()` و `nextDouble()` و `nextLong()` و `next()` (كلمة) و `nextLine()` (سطر كامل).
- **الاستيراد الصريح**: `import java.util.Scanner;` — **الاستيراد الضمني (الشامل)**: `import java.util.*;`
- اطبع دائماً **رسالة توجيهية** قبل القراءة ليعرف المستخدم ما يجب إدخاله.

Example · مثال

```
Scanner input = new Scanner(System.in);
System.out.print("Enter a radius: ");
double radius = input.nextDouble();
```

2. Identifiers

2. المعرّفات

- Names given to variables, constants, methods and classes.
- Made of letters, digits, `_` and `$`; **cannot start with a digit**.
- Cannot be a reserved word (`class`, `int`, `true` …) and can be of any length.
- Java is **case-sensitive**: `area`, `Area` and `AREA` are different. Valid: `$2`, `area`, `_x` — invalid: `2A`, `d+4`.

- أسماء نطلقها على المتغيّرات والثوابت والدوال والأصناف.
- تتكوّن من الحروف والأرقام و `_` و `$` ، و**لا يجوز أن تبدأ برقم**.
- لا يجوز أن تكون كلمة محجوزة (`class` و `int` و `true` …)، ويمكن أن تكون بأي طول.
- جافا **حسّاسة لحالة الأحرف**: `area` و `Area` و `AREA` مختلفة. صحيح: `$2` و `area` و `_x` — خطأ: `2A` و `d+4` .

3. Variables, assignment and constants — 3. المتغيّرات والإسناد والثوابت

- A **variable** stores a value that can change. Declare it with a type: `int count;`
- **Assignment** `x = 1;` stores the value on the right into the variable on the left. `x = x + 1;` is valid.
- Declare and initialise in one step: `int x = 1;`
- A **named constant** cannot change: `final double PI = 3.14159;` (UPPER_CASE names).

- **المتغيّر** يخزّن قيمة يمكن أن تتغيّر. نعرّفه مع نوعه: `int count;`
- **الإسناد** `x = 1;` يخزّن القيمة الموجودة يميناً في المتغيّر الموجود يساراً. والتعليمة `x = x + 1;` صحيحة.
- يمكن التعريف والتهيئة في خطوة واحدة: `int x = 1;`
- **الثابت المسمّى** لا تتغيّر قيمته: `final double PI = 3.14159;` (يُكتب اسمه بحروف كبيرة).

4. Numeric data types — 4. أنواع البيانات العددية

- Integers: `byte` (8-bit), `short` (16-bit), `int` (32-bit), `long` (64-bit).
- Floating-point: `float` (32-bit, ~7 digits) and `double` (64-bit, ~16 digits). `double` is the default.
- Literals: `100` is an int, `100L` a long, `5.0` a double, `5.0f` a float, `1.23456e+2` = 123.456 (scientific notation).
- Use `int` for counts and `double` for measurements with decimals.

- الأعداد الصحيحة: `byte` (8 بت) و `short` (16 بت) و `int` (32 بت) و `long` (64 بت).
- الأعداد العشرية: `float` (32 بت، نحو 7 أرقام) و `double` (64 بت، نحو 16 رقماً). والنوع الافتراضي هو `double`.
- القيم الحرفية: `100` من نوع int ، و `100L` من نوع long ، و `5.0` من نوع double ، و `5.0f` من نوع float ، و `1.23456e+2` = 123.456 (الصيغة العلمية).
- استخدم `int` للعدّ و `double` للقياسات التي فيها كسور.

5. Numeric operators — 5. العوامل الحسابية

- `+ - * / %` — addition, subtraction, multiplication, division, remainder.
- **Integer division** truncates: `5 / 2` = 2, but `5.0 / 2` = 2.5.
- **Remainder**: `7 % 3` = 1; useful for even/odd (`n % 2`), last digit (`n % 10`), and time conversions.
- Power: `Math.pow(a, b)` → `Math.pow(2, 3)` = 8.0.
- Precedence: parentheses first, then `* / %`, then `+ -`; same level evaluated left to right.

- `+ - * / %` — الجمع والطرح والضرب والقسمة وباقي القسمة.
- **القسمة الصحيحة** تحذف الكسر: `5 / 2` = 2 ، بينما `5.0 / 2` = 2.5.
- **باقي القسمة**: `7 % 3` = 1 ؛ مفيد لمعرفة الزوجي والفردي (`n % 2`) وآخر رقم (`n % 10`) وتحويل الوقت.
- الأُس: `Math.pow(a, b)` ← `Math.pow(2, 3)` = 8.0.
- الأولوية: الأقواس أولاً، ثم `* / %` ، ثم `+ -` ؛ والعوامل ذات المستوى نفسه تُحسب من اليسار إلى اليمين.

Example · مثال

```
int seconds = 500;
int minutes = seconds / 60;          // 8
int remainingSeconds = seconds % 60; // 20
double fahrenheit = 100;
double celsius = (5.0 / 9) * (fahrenheit - 32); // 37.77...
```

6. عوامل الإسناد المركّبة و ++ / -- | 6. Augmented assignment and ++ / --

- `x += 5` تعني `x = x + 5`، وكذلك `-=` و `*=` و `/=` و `%=`.
- `++i` (زيادة قبلية): أضف 1 **ثم** استخدم القيمة. `i++` (زيادة بعدية): استخدم القيمة **ثم** أضف 1.
- والأمر نفسه ينطبق على `--i` و `i--`.
- مثال: `int i = 10; int n = 10 * i++;` ← n = 100 و i = 11. ومع `++i` تكون n = 110.

- `x += 5` means `x = x + 5`; also `-=`, `*=`, `/=`, `%=`.
- `++i` (pre-increment): add 1, **then** use the value. `i++` (post-increment): use the value, **then** add 1.
- Same for `--i` and `i--`.
- Example: `int i = 10; int n = 10 * i++;` → n = 100, i = 11. With `++i`: n = 110.

7. تحويل الأنواع العددية والتحويل الصريح | 7. Numeric type conversion and casting

- في التعبير المختلط يُوسَّع النوع الأصغر تلقائياً: `int + double` ← `double`.
- التوسيع (من صغير إلى كبير) تلقائي: `double d = 3;` ← 3.0.
- التضييق (من كبير إلى صغير) يحتاج **تحويلاً صريحاً (Casting)**: `int i = (int) 3.9;` ← 3 (يُحذف الكسر).
- للاحتفاظ برقمين عشريين: `(int) (tax * 100) / 100.0`.
- في الإسناد المركّب يكون التحويل ضمنياً: `int sum = 0; sum += 4.5;` ← sum = 4.

- In a mixed expression, the smaller type is **widened** automatically: `int + double` → `double`.
- Widening (small → large) is automatic: `double d = 3;` → 3.0.
- Narrowing (large → small) needs an explicit **cast**: `int i = (int) 3.9;` → 3 (the fraction is truncated).
- Keep two decimals: `(int) (tax * 100) / 100.0`.
- In augmented assignment the cast is implicit: `int sum = 0; sum += 4.5;` → sum = 4.

8. مراحل تطوير البرمجيات | 8. Software development process

- **تحديد المتطلبات** ← **تحليل النظام** ← **تصميم النظام** ← **التنفيذ (البرمجة)** ← **الاختبار** ← **النشر** ← **الصيانة**.
- يعتمد التصميم غالباً على نمط **IPO** : **الإدخال**، ثم **المعالجة**، ثم **الإخراج**.
- مثال: برنامج قرض يُدخِل نسبة الفائدة وعدد السنوات والمبلغ، ثم يحسب القسط الشهري ويعرضه.

- **Requirements specification** → **System analysis** → **System design** → **Implementation** → **Testing** → **Deployment** → **Maintenance**.
- Design often follows the **IPO** pattern: **I**nput, **P**rocess, **O**utput.
- Example: a loan program inputs the rate, years and amount, computes the monthly payment, and displays it.

Key terms — المصطلحات الأساسية

Term	المصطلح	Meaning / المعنى
Identifier	مُعرِّف	Name of a variable, method, class or constant.
Variable	متغيّر	Named memory location whose value can change.
Constant (final)	ثابت	Value that cannot change after it is set.
Data type	نوع البيانات	int, long, double, float, byte, short …

Term	المصطلح	Meaning / المعنى
Assignment	إسناد	Storing a value: `x = 5;`
Expression	تعبير	Combination of values, variables and operators.
Integer division	القسمة الصحيحة	`/` on ints drops the fraction.
Remainder (%)	باقي القسمة	What is left after division.
Casting	التحويل الصريح للنوع	`(int) 3.9` → 3.
Literal	قيمة حرفية	A constant written directly in code: `5`, `2.5`, `100L`.
IPO	إدخال – معالجة – إخراج	Input, Process, Output design pattern.

Part 2 — Quiz

الجزء الثاني — الاختبار

Choose the **one** correct answer for each question (3 options). Answers are on the last page.	اختر الإجابة **الصحيحة الوحيدة** لكل سؤال (3 خيارات). الإجابات في الصفحة الأخيرة.
Name: _____ Score: ____ / 15	الاسم: _____ الدرجة: _____ / 15

Q1. Which identifier is **valid**?

- a) `2ndValue`
- b) `total_Sum`
- c) `class`

س (1) أي معرّف مما يلي **صحيح**؟

- أ) `2ndValue`
- ب) `total_Sum`
- ج) `class`

Q2. Which statement reads a decimal number from the keyboard?

- a) `double d = input.nextDouble();`
- b) `double d = input.nextInt();`
- c) `double d = input.next();`

س (2) أي تعليمة تقرأ عدداً عشرياً من لوحة المفاتيح؟

- أ) `double d = input.nextDouble();`
- ب) `double d = input.nextInt();`
- ج) `double d = input.next();`

Q3. What is the value of `17 / 5`?

- a) `3.4`
- b) `4`
- c) `3`

س (3) ما قيمة `17 / 5` ؟

- أ) `3.4`
- ب) `4`
- ج) `3`

Q4. What is the value of `17 % 5`?

- a) `2`
- b) `3`
- c) `3.4`

س (4) ما قيمة `17 % 5` ؟

- أ) `2`
- ب) `3`
- ج) `3.4`

Q5. How do you declare a constant for PI?

- a) `const double PI = 3.14159;`
- b) `final double PI = 3.14159;`
- c) `static PI = 3.14159;`

س (5) كيف نعرّف ثابتاً لقيمة PI ؟

- أ) `const double PI = 3.14159;`
- ب) `final double PI = 3.14159;`
- ج) `static PI = 3.14159;`

Q6. What is the result of `Math.pow(2, 3)`?

- a) `6`
- b) `9.0`
- c) `8.0`

س (6) ما ناتج `Math.pow(2, 3)` ؟

- أ) `6`
- ب) `9.0`
- ج) `8.0`

Q7. Evaluate `3 + 4 * 2 - 6 / 3`.

- a) `9`
- b) `12`
- c) `4`

س (7) احسب قيمة `3 + 4 * 2 - 6 / 3` .

- أ) `9`
- ب) `12`
- ج) `4`

Q8. After `int i = 5; int j = i++;` what are i and j?

- a) i = 6, j = 6
- b) i = 6, j = 5
- c) i = 5, j = 6

س (8) بعد تنفيذ `int i = 5; int j = i++;` ما قيمتا i و j ؟

- أ) i = 6 و j = 6
- ب) i = 6 و j = 5
- ج) i = 5 و j = 6

Q9. After `int x = 10; x -= 3; x *= 2;` what is x?

- a) `14`
- b) `17`
- c) `4`

س (9) بعد `int x = 10; x -= 3; x *= 2;` ما قيمة x ؟

- أ) `14`
- ب) `17`
- ج) `4`

Q10. What is the value of `(int) 7.9`?

- a) `8`
- b) `7.0`
- c) `7`

س (10) ما قيمة `(int) 7.9` ؟

- أ) `8`
- ب) `7.0`
- ج) `7`

Q11. What is the type of the expression `5 + 2.0`?

- a) `int`
- b) `double`
- c) `long`

س (11) ما نوع التعبير `5 + 2.0` ؟

- أ) `int`
- ب) `double`
- ج) `long`

Q12. Which statement causes a compile error?

- a) `double d = 5;`
- b) `long n = 5;`
- c) `int i = 5.5;`

س (12) أي تعليمة تسبّب خطأ ترجمة؟

- أ) `double d = 5;`
- ب) `long n = 5;`
- ج) `int i = 5.5;`

Q13. What does `1.5e2` represent?

- a) `150.0`
- b) `1.52`
- c) `0.015`

س (13) ماذا تمثّل القيمة `1.5e2` ؟

- أ) `150.0`
- ب) `1.52`
- ج) `0.015`

Q14. What does the IPO pattern stand for?

- a) Input, Process, Output
- b) Import, Print, Operate
- c) Integer, Precision, Operator

س (14) إلام يشير نمط IPO ؟

- أ) إدخال، معالجة، إخراج
- ب) استيراد، طباعة، تشغيل
- ج) عدد صحيح، دقّة، عامل

Q15. Which phase comes **right after** implementation in the software development process?

- a) Deployment
- b) Testing
- c) System analysis

س (15) أي مرحلة تأتي **مباشرة بعد** التنفيذ في عملية تطوير البرمجيات؟

- أ) النشر
- ب) الاختبار
- ج) تحليل النظام

Answer key

مفتاح الإجابات

#	Answer · الإجابة	Why	التوضيح
1	**b) / ب** `total_Sum`	An identifier cannot start with a digit or be a reserved word.	لا يجوز أن يبدأ المعرّف برقم أو أن يكون كلمة محجوزة.
2	**a) / أ** `double d = input.nextDouble();`	`nextDouble()` reads a double; `next()` returns a String.	`nextDouble()` تقرأ عدداً عشرياً، أما `next()` فتعيد نصاً.
3	**c) / ج** `3`	Both operands are int → integer division truncates.	كلا الطرفين من نوع int ← القسمة الصحيحة تحذف الكسر.
4	**a) / أ** `2`	17 = 5 × 3 + 2.	17 = 5 × 3 + 2.
5	**b) / ب** `final double PI = 3.14159;`	Java uses the keyword `final` for constants.	تستخدم جافا الكلمة المحجوزة `final` لتعريف الثوابت.
6	**c) / ج** `8.0`	$2^3 = 8$, and `pow` returns a double.	2 أس 3 = 8 ، والدالة `pow` تعيد قيمة من نوع double.
7	**a) / أ** `9`	`4*2` = 8 and `6/3` = 2 first, then 3 + 8 − 2 = 9.	نحسب أولاً `4*2` = 8 و `6/3` = 2 ، ثم 3 + 8 − 2 = 9.
8	**b) / ب** i = 6, j = 5	Post-increment: j gets the old value 5, then i becomes 6.	الزيادة البعدية: تأخذ j القيمة القديمة 5 ، ثم تصبح i تساوي 6.
9	**a) / أ** `14`	10 − 3 = 7, then 7 × 2 = 14.	10 − 3 = 7 ، ثم 7 × 2 = 14.
10	**c) / ج** `7`	Casting to int truncates the fraction (it does not round).	التحويل إلى int يحذف الكسر (لا يقرّب).
11	**b) / ب** `double`	The int is widened to double, so the result is 7.0 (double).	يُوسَّع الـ int إلى double ، فتكون النتيجة 7.0 من نوع double.
12	**c) / ج** `int i = 5.5;`	Assigning a double to an int is narrowing and needs a cast.	إسناد double إلى int تضييق ويحتاج إلى تحويل صريح.
13	**a) / أ** `150.0`	Scientific notation: $1.5 \times 10^2 = 150.0$.	الصيغة العلمية: $1.5 \times 10^2 = 150.0$.
14	**a) / أ** Input, Process, Output	Most simple programs read input, process it, and display output.	معظم البرامج البسيطة تقرأ المدخلات وتعالجها ثم تعرض المخرجات.
15	**b) / ب** Testing	Requirements → analysis → design → implementation → **testing** → deployment → maintenance.	المتطلبات ← التحليل ← التصميم ← التنفيذ ← **الاختبار** ← النشر ← الصيانة.