







# C++

## CHAPTER 3: INPUT/OUTPUT

## Contents

 Chapter 3: Input / Output .....	3
cin (int).....	6
Cin(double) .....	6
<b>Cin(char)</b> .....	7
cin(string).....	7
cin(bool) .....	8
cin.get().....	8
getline(cin, string) .....	10
Input Failure  .....	12
<b>!</b> Input Failure .....	16
أسئلة اختيار من متعدد (C++ Input Mixing) .....	19
 <iomanip> التحكم بشكل الطباعة (Formatting).....	22
 <fstream> C++ التعامل مع الملفات في .....	26
test bank chapter 3 .....	31
Chapter 3 – Input/Output and File Handling — Practice Questions .....	38

## Chapter 3: Input / Output

⚙️ نتذكر شويث شغلات :

في لغة C++ ، أي تعامل بين المستخدم والبرنامج، سواء طباعة أو إدخال بيانات، سيتم من خلال مكتبة اسمها **<iostream>**.

هاي المكتبة فيها أهم أداتين للتعامل مع الإدخال والإخراج:

- **cout** (Output) تستخدم لعرض المعلومات على الشاشة ←
- **cin** (Input) تستخدم لقراءة المعلومات من المستخدم ←

**cout -** الطباعة على الشاشة: 🖨️

```
#include <iostream>
using namespace std;
```

```
int main() {
    cout << "أهلاً بك في C++!" << endl;
    int x = 5;
    cout << "x : " << x << endl;

    return 0;
}
```

💡 ملاحظات:

- العلامة << معناها "أرسل إلى الشاشة".
- endl معناها "انزل لسطر جديد".
- ممكن تطبع أكثر من إشي بنفس السطر:

**cin -** قراءة من المستخدم:

الكود الاتي بطلب من المستخدم يدخل عمره وبيطبعه:

```
#include <iostream>
using namespace std;
```

```
int main() {
    int age;
    cout << "كم عمرك؟";
    cin >> age;
    cout << age << endl;
    return 0;
}
```

```
int a, b;
cin >> a >> b;
```

ملاحظات:

- العلامة >> معناها "أُخذ من المستخدم".
- ممكن تقرأ أكثر من قيمة وراء بعض:

نقاط مهمة لازم تنتبه لها:

1. الفراغات ما بتأثر على cin و cout، بس:
  - cin بتقرأ أول كلمة فقط، ولو أدخل المستخدم سطر فيه فراغات، رح تأخذ أول كلمة فقط وتكمل.
2. قبل تعيين أي قيمة لمتغير، ممكن يكون فيه داخله "rubbish data" يعني بيانات عشوائية ملهاش معنى.
3. الـ **char** بتقدر تخزن رمز واحد، وبتتعامل مع الرموز الخاصة (زي \n, \t) كأنها حرف واحد.
4. الـ **string** بتقدر تحتوي على رموز خاصة + كلمات بشكل طبيعي:

```
string s = "Hello\nWorld";
cout << s;
// رح يطبع:
// Hello
// World
```

مثال للتذكير: 


```
#include <iostream>

using namespace std;

int main() {
    int id;
    double gpa;
    char grade;

    cout << "ادخل رقمك الجامعي، المعدل، والتقدير ";
    cin >> id >> gpa >> grade;

    cout << "الرقم: " << id << ", المعدل: " << gpa << ", التقدير: " << grade << endl;
    return 0;
}
```

تنويه مهم: 

- لو ما استخدمت using namespace std; لازم تكتب std::cin و std::cout بدل cin و cout.

## cin (int)

التفاصيل	البند
عند استخدام cin مع متغير من نوع int، يتوقع المُدخل يكون رقمًا صحيحًا فقط (موجب أو سالب). أي فاصلة عشرية أو حرف تعتبر نهاية للقراءة. إذا دخل المستخدم قيمة غير صالحة (زي حرف)، بتصير "Input failure"	الشرح
int x; cin >> x;	الكود
Stored Value	Input
46825	46825
23	23
	228.5 (توقّف عند الفاصلة) 228
	123 45 (الأول فقط) 123
-7	-7 10 2023
Input failure	A 48

## Cin(double)

التفاصيل	البند
double يشبه int، لكن يقبل الفاصلة العشرية. إذا دخل المستخدم رقم عشري، يتم حفظه كما هو. لو دخل المستخدم رقم صحيح، يتم حفظه كـ double يعني 10.0	الشرح
double x; cin >> x;	الكود
Stored Value	Input
46825.0	46825
0.23	0.23
0.67	0.67

## Cin(char)

التفاصيل	البند
char يخزن حرف واحد فقط. عند استخدام cin، يتم قراءة أول رمز فقط من المدخل (رقم، حرف، أو رمز خاص). لا يتم قبول space.	الشرح
cpp char x; cin >> x;	الكود
Stored Value	Input
'4'	46825
't'	true2s3
'@'	@hello

## cin(string)

التفاصيل	البند
string تقرأ سلسلة أحرف بدون فراغات. تتوقف القراءة عند أول space أو newline. إذا احتجت قراءة جملة كاملة فيها فراغات، لازم تستخدم getline().	الشرح
cin >> x; string x;	الكود
Stored Value	Input
"46825"	46825
"s"	sara
"Intro"	Intro to pr

## cin(bool)

التفاصيل	البند
bool يقبل فقط 1 (true) أو 0 (false). بعض الكومبايلرات بتقبل كلمات زي true و false. أي إدخال غير مفهوم ممكن يعطي "Input failure" أو يحوله حسب نوع الكومبايلر.	الشرح
cin >> x;	الكود bool x;
Stored Value	Input
true	1
false	0
true	إذا الكومبايلر بيدعمه (true)
false	أحياناً (false)
غير معروف (يعتمد على الكومبايلر)	A

## cin.get()

لما نستخدم `cin >> ch;` لقراءة حرف، فهي تقرأ أول **character** عادي (رقم، حرف، رمز)، لكنها بتتجاهل الفراغات والمسافات (**spaces**) والسطر الجديد (**Enter / newline**).  
 يعني إذا المستخدم دخل `space` أو `كيس Enter`، `cin` العادية ما رح تقرأهم. ❌

الحل: `cin.get(ch);` ✅

بتخلينا نقرأ أي شيء حرفياً، حتى لو كان:

- Space ' '
- New line \n
- Tab
- أو أي رمز ثاني



مثال توضيحي: 🔍

```
char ch;  
cin.get(ch);
```

القيمة المخزنة داخل ch	input
'A'	A
' '	(فراغ)
'\n'	Enter
'9'	9

مثال عملي متسلسل: 💡

```
char x, y, z;  
cin >> x >> y;  
cin.get(z);
```

إذا المستخدم كتب:


IT←

القيمة	المتغير
'I'	x
'T'	y
← '\n' لأنه cin.get() قرأت الـ Enter	z

ملاحظات مهمة: ⚠️

- cin.get() لازم تستخدم مع char فقط.
- مفيدة جدًا لما تحتاج تقرأ حرف من مدخل فيه فراغات.

## getline(cin, string)

المشكلة مع cin >> string 

لما نستخدم:

```
string x;  
cin >> x;
```


الـ cin بتقرأ كلمة وحدة فقط، ويتوقف عند أول space أو Enter، يعني:

إذا كتب المستخدم:

Intro to programming

فرح يخزن في x فقط:

"Intro"

الباقي "to programming" ما بينقرأ. 


---

✓ الحل: getline(cin, x)

getline بتقرأ سطر كامل، وبتخزن كل إشي فيه حتى لو كان فيه فراغات أو رموز.

وتوقف فقط لما المستخدم يضغط Enter.

---

مثال بسيط: 

```
string s;  
getline(cin, s);
```

المستخدم دخل	القيمة المخزنة داخل s
Intro to pr	"Intro to pr"
Hello	"Hello"
A sentence with spaces	"A sentence with spaces"

⚠ ملاحظات مهمة:

**getline()** تستخدم فقط مع متغير من نوع **string**.

إذا استخدمنا `cin >>` قبل `getline()`، ممكن يحصل خطأ أو يتخزن سطر فاضي!

🧠 مثال مشترك يوضح المشكلة:

```
char ch;
string s;
cin >> ch;
getline(cin, s);
```

إذا المستخدم دخل:

```
3
Jordan
```

المتغير	القيمة
ch	'3'
s	/n

🔧 السبب: `cin >> ch` قرأت 3، وبعدها ظل ال\n، و `getline` قرأته على إنه بداية السطر الجديد.

✂ الحل :

cin.get(a);

أو

cin.ignore();

يتجاوز الـ newline المتروك

---

## Input Failure

أولاً: شو يعني Input Failure ؟

لما تستخدم >>cin مع متغير، لازم المستخدم يدخل نفس نوع البيانات.  
لو أدخل نوع مختلف، بصير فشل بالقراءة، ويتسمى:

### Input Failure

✂ شو بصير لما يصير فشل بالإدخال؟

1. cin يقف قراءة، وكل متغير بعده ما بيتخزن فيه إشي.
2. المتغير اللي صار فيه المشكلة بضل فيه قيمة قديمة أو 0 أو rubbish.
3. لو ما عالجنا الخطأ، البرنامج بيضل يمشي كأنو مافي إشي غلط!

✓ قبل ما نحكي كيف نحل المشكلة، خلينا نشوف هذا المثال المهم:

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    int age;
```

```
char grade;

string fullName;

cout << "ادخل عمرك";
cin >> age;

cout << "ادخل تقديرك (حرف)";
cin >> grade;

cout << "ادخل اسمك الكامل";
getline(cin, fullName); // هون في مشكلة ❌

cout << "\n----- البيانات ----- \n";
cout << "العمر: " << age << endl;
cout << "التقدير: " << grade << endl;
cout << "الاسم: " << fullName << endl;

return 0;
}
```

شو المشكلة هون؟	لاحظ أن الاسم طلع فارغ رغم إنك كتبتة 😊 !
البرنامج بيطلع إشي زي:	ليش؟
ادخل عمرك: 20	لأنه بعد cin >> grade; يفضل فيه newline
ادخل تقديرك (حرف) A :	(نتيجة ضغط المستخدم على Enter ،
ادخل اسمك الكامل :	و getline اقرأ هذا newline ، وظن إنه نهاية السطر، فخزن سطر فاضي.

العمر: 20	الحل باستخدام <code>cin.get()</code>
التقدير: A	قبل استخدام <code>getline()</code> ، لازم نمسح السطر الفاضي:
الاسم:	<code>cin.get(); //</code> وبيقرأ ويمسح ال <code>newline</code> المتروك
	<code>getline(cin, fullName);</code>

الكود الصحيح بعد التعديل: 

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    int age;
    char grade;
    string fullName;
    char x;
    cout << "ادخل عمرك ";
    cin >> age;

    cout << "ادخل تقديرك (حرف) ";
    cin >> grade;

    cin.get(x);
```

```

cout << "ادخل اسمك الكامل ";
getline(cin, fullName); // الآن تقرأ الاسم كامل

cout << "\n----- البيانات ----- \n";
cout << "العمر: " << age << endl;
cout << "التقدير: " << grade << endl;
cout << "الاسم: " << fullName << endl;

return 0;
}

```

👉 شو بنعمل بعد هيك؟

بنا نطور هذا المثال شوي ونضيف فيه إدخال خطأ (مثلاً يدخل حرف بدل العمر)، ونشوف:

- كيف تصير **Input Failure**
- كيف نعرف إنه صار فشل؟
- كيف نحله باستخدام `cin.clear()` و `cin.ignore()`

يمكن نستخدم بدل `cin.get(char)` دالة اسمها `cin.ignore()`، وهي مفيدة عشان نتجاوز أي حرف متروك في السطر زي الـ Enter قبل ما نستخدم `getline()`، وخصوصاً لما نكون استعملنا `>>cin` قبلها.

```

cin.ignore();
getline(cin, fullName);

```

✔ بتشتغل تقريباً زي `cin.get()` في هيك حالات، وبنستخدمها كتير لما بدنا نكمل قراءة السطر كامل بدون مشاكل.

**فشل الإدخال في C++**

## ! Input Failure

لما البرنامج يتوقع المستخدم يدخل نوع بيانات معين، زي رقم int أو double لكن المستخدم بيدخل إشي مش من نفس النوع، مثل:

- يدخل حرف بدل رقم
- يدخل كلمة بدل رقم
- أو يحط رمز مش مناسب

البرنامج ما بقدر يخزن القيمة، ويبصير فيه إشي اسمه:

### 💣 Input Failure (فشل في الإدخال)

مثال بسيط: 🖋️

```
int age;  
cin >> age;
```

لو المستخدم كتب:

Nginx

او

twenty

بدل ما يكتب رقم زي 20، رح بصير فشل، ويبصير التالي:

- ما بتخزن القيمة داخل age
- cin بيوقف وما بقرأ أي قيم بعدها
- وباقي المتغيرات اللي بعده رح يتم تجاهل إدخالها



## شو بيصير في المتغيرات وقت الفشل؟

نوع المتغير	في حالة Input Failure
int	بيصير 0 أو قيمة عشوائية
double	0.0 أو rubbish
bool	غالبًا false
char	ممكن يحتفظ بالقيمة القديمة أو قيمة غير مفهومة
string	ما بتتأثر، بتمشي طبيعي

✓ مثال جديد لتوضيح المشكلة:

```
#include <iostream>
#include <string>
using namespace std;

int quantity;
double cost;
string item;

int main() {
    cout << "كمية المنتج ";
    cin >> quantity;

    cout << "سعر المنتج ";
    cin >> cost;

    cout << "اسم المنتج ";
```

```
cin >> item;
```

```
cout << "\n--- ملخص ---\n";
```

```
cout << "الكمية: " << quantity << endl;
```

```
cout << "السعر: " << cost << endl;
```

```
cout << "الاسم: " << item << endl;
```

```
return 0;
```

```
}
```

المستخدم كتب: 🖥️👤

ten

8.99

Apple

⚠️ شو رح يصير؟

- البرنامج حاول يخزن ten داخل quantity وهو → int فشل
- تجاهل إدخال cost و item

• الناتج رح يكون:

الكمية: 0

السعر: 0

الاسم:

📌 ملاحظات مهمة :

- cin كلما يفشل، بوقف وما بأكمل.
- المتغير اللي صار فيه الخطأ ما بتخزن فيه القيمة الصح.

- المتغيرات التي بعده ما رح يتم إدخالها نهائياً.
- هذا النوع من الأخطاء صعب يبين، ولأزم ننتبه إله من خلال تجربة البرنامج.

## (C++ Input Mixing) أسئلة اختيار من متعدد

What is the output of the following code?

السؤال الأول

```
#include <iostream>
#include <string>
using namespace std;
int main() {
    int a, b;
    string name;

    cin >> a >> b;

    getline(cin, name);


    cout << a << b << name;

    return 0;
}
```


Input :

- 10 20
- Ali Ahmad

- A) 1020Ali Ahmad
- B) 1020
- C) 1020 Ahmad
- D) 1020 Ali
- E) 10 20 Ali Ahmad

الإجابة B: 

لأن `getline` تقرأ السطر الفاضي بعد `cin`

 السؤال الثاني

```
#include <iostream>

using namespace std;
```

```
int main() {
double num;

char op;

cin >> num;

cin.get(op);

cout << num << op;

return 0;}
```


Input:

- 34.5+

- A) 34.5+
- B) 34.5
- C) 34.5 (مسافة)
- D) 0+
- E) +34.5

الإجابة C: ☒

لأن cin.get(op) تقرأ الفراغ بعد 34.5

 السؤال الثالث: float + int + string

```
#include <iostream>

#include <string>

using namespace std;
```

```
int main(){
float x;

int y;

string word;

cin >> x >> y;

getline(cin, word);
```

Input :

- 12.5 7
- Hello World

```
cout << x << y << word;  
return 0;  
}
```

- A) 12.57Hello
- B) 12.57Hello World
- C) 12.57
- D) 12.57 World
- E) 12.5 7 Hello World

الإجابة C: ☒ .  
getline يتقرأ السطر الفاضي بعد الـ cin

#### السؤال الرابع

```
#include <iostream>  
#include <string>  
using namespace std;  
int main() {  
    int age;  
    string hobby;  
    char grade;  
    cin >> age;  
    getline(cin, hobby);  
    cin >> grade;  
    cout << age << hobby << grade;  
    return 0;  
}
```

Input;

- 22
- Football
- A

- A) 22FootballA
- B) 22A
- C) 22 FootballA
- D) 22AFootball
- E) 22AFootball

الإجابة B: ☒ .  
getline يتقرأ الفراغ بعد Enter ، ف hobby سطر فاضي

#### السؤال الخامس

```
#include <iostream>  
#include <string>  
using namespace std;  
int main() {  
    int id;  
    string line1, line2;  
    cin >> id;
```

Input:

- 1023
- This is line one
- And this is line two

```
getline(cin, line1);
getline(cin, line2);
cout << id << line1 << line2;
return 0;
}
```

- A) 1023This is line two
- B) 1023This is line oneAnd this is line two
- C) 1023And this is line two
- D) 1023 This is line oneAnd this is line two
- E) 1023This is lineoneAndthisislinetwo

الإجابة A: 

line1 يتخزن السطر الفاضي، line2 يتخزن "This is line one"

## <iomanip> التحكم بشكل الطباعة (Formatting)

حتى نستخدم أدوات التنسيق في الإخراج، لازم نضيف المكتبة التالية في أعلى الكود:

```
#include <iomanip>
```

### 1. showpoint

◆ بتجبر البرنامج إنه يطبع الفاصلة العشرية والأصفار بعدها حتى لو ما كان فيه كسور فعلاً.

◆ بتأثر على الأنواع العشرية مثل float و double.

مثال: 

```
#include <iostream>
#include <iomanip>
using namespace std;

int main() {
    double x = 3;
    cout << showpoint << x;
```

Output:

3.00000

```
return 0;  
}
```

حتى لو كانت عدد صحيح، تم طباعته بـ 5 خانات عشرية بسبب showpoint.

## ✳ 2. setprecision(n)

- ◆ بتحدد عدد الخانات الظاهرة بعد أو قبل الفاصلة العشرية.
- ◆ إذا ما استخدمنا fixed، رح يؤثر على عدد الخانات الإجمالي (قبل + بعد الفاصلة).

مثال: 

```
#include <iostream>  
#include <iomanip>  
using namespace std;  
int main() {  
    double a = 9.876543;  
    cout << showpoint << setprecision(4) << a;  
    return 0;}
```

Output:

9.877

💡 العدد الإجمالي 4 خانات، والرقم تم تقريبه.

## ✳ 3. fixed

- ◆ لما نستخدمها مع setprecision(n)، بتجبر البرنامج يطبع n خانات بعد الفاصلة فقط، بغض النظر عن العدد قبل الفاصلة.

مثال: 

```
#include <iostream>  
#include <iomanip>  
using namespace std;
```

Output:

123.457

```
int main() {  
    double salary = 123.45678;  
    cout << fixed << showpoint << setprecision(3) << salary;  
    return 0;};
```

تم تقريب العدد إلى 3 خانات بعد الفاصلة فقط، وتم عرض الفاصلة بـ showpoint.

#### 🔦 ملاحظات مهمة:

- showpoint لئلا ما يتحكم بعدد الخانات، بس بتجبر الطباعة بالفاصلة.
- setprecision بدون fixed بتأثر على كامل الرقم (قبل وبعد الفاصلة).
- fixed + setprecision بتحدد عدد الخانات بعد الفاصلة فقط.
- كل الأوامر هاي بتكون مؤثرة فقط على القيم العشرية (float, double) وما إلها تأثير على int.

#### 🔧 Question 1: What is the output of the following code?

```
#include <iostream>  
#include <iomanip>  
using namespace std;  
int main() {  
    double x = 7;  
    cout << showpoint << x;  
    return 0;  
}
```



- A) 7
- B) 7.00000
- C) 7.00
- D) 7.000
- E) Error

 **Correct Answer: B**

---

 **Question 2: What is the output of the following code?**

```
#include <iostream>
#include <iomanip>
using namespace std;

int main() {
    double grade = 89.6789;
    cout << fixed << showpoint << setprecision(2) << grade;
    return 0;
}
```

- A) 89.67
- B) 89.68
- C) 89.6789
- D) 89.6
- E) 89.00

 **Correct Answer: B**

## <fstream> التعامل مع الملفات في C++

ببساطة: بدل ما نستخدم cin و cout لإدخال وإخراج البيانات من الشاشة،  
نستخدم ifstream و ofstream للقراءة والطباعة من ملفات نصية.

✂ المكتبة المطلوبة

```
#include <fstream>
```

✱ أنواع الملفات في C++

يشبه...	الاستخدام	النوع
cin	للقراءة من ملف	ifstream
cout	للكتاباة/الطباعة إلى ملف	ofstream

📁 خطوات العمل مع الملفات

1. ✅ تجهيز المكتبة:

```
#include <fstream>
```

2. ✅ إنشاء متغير للملف:

```
ifstream inFile; // للقراءة  
ofstream outFile; // للطباعة
```

3. ✅ فتح الملف:

```
inFile.open("input.txt");  
outFile.open("output.txt");
```

- ◆ إذا كان الملف موجود: البرنامج يقرأ أو يكتب عليه.
- ◆ إذا مش موجود ofstream: بعمله تلقائي، لكن ifstream ما يشتغل إلا إذا الملف موجود فعلاً.

✍ مثال عملي: قراءة وطباعة من/إلى ملف

نفترض أن ملف input.txt يحتوي على:

Mohammad

20

الكود: 

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

int main() {
    ifstream in;
    ofstream out;

    in.open("input.txt");
    out.open("output.txt");

    string name;
    int age;

    getline(in, name);
    in >> age;
```

```

out << name << endl;

out << age;


in.close();

out.close();


return 0;
}

```

◆ هذا الكود:

- يقرأ name من السطر الأول
- يقرأ age من السطر الثاني
- ييطبعهم على ملف output.txt

🔴 ملاحظات مهمة:

1. لازم تغلق الملفات بعد ما تخلص باستخدام:

```

in.close();

out.close();

```

2. تقدر تقرأ وتكتب باستخدام نفس المتغير (fstream) إذا بدك تستخدم الاثنين مع بعض، لكن مش ضمن المنهاج.

3. لازم يكون الملف موجود في نفس مجلد المشروع أو توفّر المسار الصحيح).

✎ مصطلحات مهمة

المصطلح	الشرح البسيط
in.open(...)	فتح ملف للقراءة
out.open(...)	فتح ملف للكتابة (أو إنشاءه)

<b>getline(in, x)</b>	قراءة سطر كامل من ملف
<b>in &gt;&gt; x</b>	قراءة كلمة أو رقم
<b>out &lt;&lt; x</b>	طباعة على الملف
<b>close()</b>	إغلاق الملف بعد الانتهاء

### Question: File Reading and Writing

cpp

نسخة تحرير

```
#include <iostream>
```

```
#include <fstream>
```

```
#include <string>
```

```
using namespace std;
```

```
int main() {
```

```
    ifstream inFile;
```

```
    ofstream outFile;
```

```
    string name;
```

```
    int age;
```

```
    inFile.open("input.txt");
```

```
    outFile.open("output.txt");
```

```
    getline(inFile, name);
```

```
    inFile >> age;
```

```
outFile << name << age;
```

```
inFile.close();
```

```
outFile.close();
```

```
return 0;
```

```
}*****  
*
```

**Assume the content of input.txt is:**

Ali Ahmad

22

**What is the output that will be stored inside output.txt?**

A) Ali Ahmad

22

B) Ali

22

C) Ali Ahmad22

D) Ali22

E) Error: file not found



**Correct Answer: C**

**Explanation:**

- getline(inFile, name) will read the full line → name = "Ali Ahmad"
- inFile >> age will read the number 22
- outFile << name << age will write them **in one line without space** → "Ali Ahmad22"

## test bank chapter 3

1-What is the program output?

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    int p, q;
    string s;
    cin >> p >> q;
    getline(cin, s);
    cout << p << q << "|" << s;
    return 0;
}
```

**Input:**

5 8

Programming C++

- A) 58|Programming C++
- B) 58| C++
- C) 58|
- D) 5 8|Programming C++
- E) Error

👍 **Correct: C** (the leftover newline causes empty s)

## 2-What's printed?

```
#include <iostream>
using namespace std;

int main(){
double d; char ch;

    cin >> d;

    cin.get(ch);

    cout << d << "/" << (int)ch;

    return 0;
}
```

**Input:**

2.5A

- A) 2.5/a
- B) 2.5/A
- C) 2.5/0
- D) 2.5/65
- E) Error

✅ **Correct: D** (ch is newline ASCII 65 after reading 'A', so likely 65)

## 3-What is the program output?

```
#include <iostream>
#include <iomanip>
using namespace std;
```



```
int main(){  
    double x = 123.4567;  
    cout << showpoint << setprecision(4) << x;  
    return 0;  
}
```

- A) 123.5
- B) 123.4567
- C) 1.235e+02
- D) 123.5?
- E) 123.457

☒ **Correct: A** (precision=4 total digits → "123.5")

---

4- What is the program output?

```
#include <iostream>  
#include <iomanip>  
using namespace std;  
int main(){  
    double n = 9.9999;  
    cout << fixed << setprecision(2) << n;  
    return 0;  
}
```

- A) 10.00
- B) 9.99
- C) 9.9999
- D) 9.100
- E) 9.99?

☒ **Correct: A** (rounded to two decimals)

---

**5- What is the program output?**

Suppose input.txt contains:

Beta Tester

30

```
#include <fstream>
#include <iostream>
#include <string>
using namespace std;

int main(){
    ifstream in("input.txt");
    ofstream out("result.txt");
    string name;
    int age;
    getline(in, name);
    in >> age;
    out << age << " " << name;
    return 0;
}
```

What's the content of result.txt?

- A) 30 Beta Tester
- B) Beta Tester 30
- C) 30  
Beta Tester
- D) Error
- E) (empty)

✓ **Correct: A**

6- What is the program output?

Which header and class is correct for reading from a file?

- A) <ifstream> and ofstream
- B) <fstream> and ifstream
- C) <iostream> and fstream
- D) <fstream> and ofstream
- E) <fstream> and fstream

✓ **Correct: B**

[en.wikipedia.org+13geeksforgeeks.org+13quizizz.com+13scribd.com+7quizizz.com+7quizlet.com+7quizizz.com+6sanfoundry.com+6quizlet.com+6quizizz.com+3quizlet.com+3tutorialspoint.com+3scribd.com+1coursehero.com+1stackoverflow.com+3scribd.com+4sanfoundry.com+4cplusplus.com+4](#)

7- What is the program output?

**Input:**

x 9.8

```
#include <iostream>

using namespace std;

int main(){
    int a;
```

```
double b;  
  
cin >> a >> b;  
  
cout << a << " " << b;  
  
return 0;  
  
}
```

What's printed?

- A) 0 9.8
- B) garbage garbage
- C) 0 0
- D) Error
- E) 0 (ignored)

☒ **Correct: C** (failure stops before reading b → default zero)

8- What is the program output?

**Input:**

Z Hello World

```
#include <iostream>  
  
#include <string>  
  
using namespace std;  
  
int main(){  
    char c;  
    string line;  
  
    cin >> c;  
  
    getline(cin, line);  
  
    cout << c << "|" << line;  
  
    return 0;  
  
}
```

Output is:

- A) Z|Hello World
- B) Z| World
- C) Z||Hello World
- D) Z| Hello World
- E) Error

✅ **Correct: D** (line starts with the space)

---

9-What is the program output?

```
#include <iostream>
#include <iomanip>
using namespace std;

int main(){
    double v = 1.2;
    cout << v << " ";
    cout << showpoint << v << " ";
    cout << fixed << setprecision(4) << v;
    return 0;}
```

Correct output?

- A) 1.2 1.20000 1.2000
- B) 1.2 1.2 1.2000
- C) 1.2 1.20000 1.2000
- D) 1.2 1.20000 1.2
- E) 1.2 1.2 1.2

✅ **Correct: C**

---

## Chapter 3 – Input/Output and File Handling — Practice Questions

---

### Question 1 — Mixing cin with cin.get

```
#include <iostream>
using namespace std;

int main(){
    char x, y;
    int num;
    cin >> x >> num;
    cin.get(y);
    cout << x << "-" << num << "-" << int(y);
    return 0;
}
```

**Input:**

A 100#

**What is the output?**

A) A-100-35

B) A-100-10

C) A-100-32

D) A-100-0

E) A-100-#

**Correct answer: B**

---

### Question 2 — getline after reading int

```
#include <iostream>
```

```
#include <string>

using namespace std;

int main(){

    int id;

    string comment;

    cin >> id;

    getline(cin, comment);

    cout << id << "-" << comment.length();

    return 0;

}
```

**Input:**

55

Nice course!

**What is the output?**

- A) 55-12
- B) 55-13
- C) 55-0
- D) 55-1
- E) 55-11

**Correct answer: C**

---

**Question 3 — setprecision without fixed**

```
#include <iostream>

#include <iomanip>

using namespace std;
```

```
int main(){  
    double x = 987.6543;  
    cout << showpoint << setprecision(3) << x;  
    return 0;  
}
```

**What is the output?**

- A) 987.654
- B) 987.65
- C) 9.88e+02
- D) 988
- E) 987.6

**Correct answer: D**

---

**Question 4 — Writing to a file with ofstream**

```
#include <fstream>  
using namespace std;  
  
int main(){  
    ofstream out("data.txt");  
    int age = 30;  
    out << "Age: " << age;  
    out.close();  
    return 0;  
}
```

**What will be the content of data.txt?**

- A) Age 30
- B) Age: 30
- C) 30



- D) "Age: 30"
- E) Error

**Correct answer: B**

---

#### Question 5 — Input Failure Handling

```
#include <iostream>
using namespace std;
int main(){
    int x;
    cin >> x;
    cout << x + 5;
    return 0;
}
```

**Input:**

hi

**What is the output?**

- A) 5
- B) 0
- C) Error
- D) Garbage
- E) Nothing

**Correct answer: A**