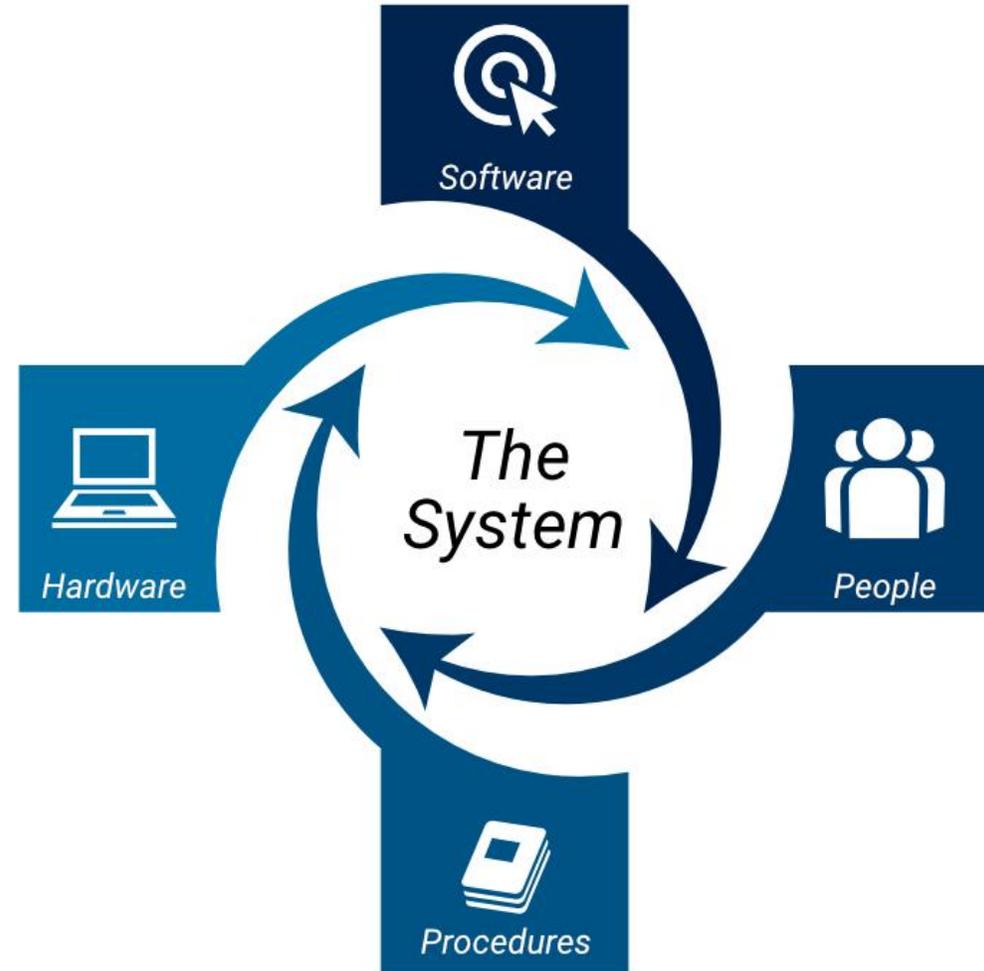


SYSTEM TESTING

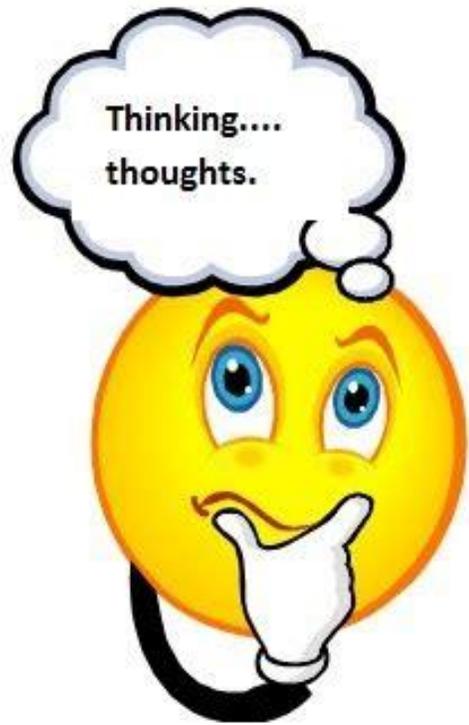


Objectives

At the end of the lesson students should be able to:

- State what is system testing
- Give reasons for system testing
- Describe normal test data
- Describe extreme test data
- Describe erroneous test data
- Describe boundary test data
- Construct a test plan for data testing





Why do you test/taste things/food
before using or eating them?

Discuss this with your classmates and
compare answers.

Make a record of your answer

System testing

A level of software testing that **evaluates the complete**, integrated software system to verify that it meets the specified **functional**(what) and **non-functional** (specify how well) requirements.

Reasons for System Testing

Holistic Validation

Requirement Compliance.

Defect Detection

Risk Mitigation



Execution Errors

Execution errors at program startup refer to **runtime exceptions or failures** that occur during the initial loading and initialization phase of a software program, right after it begins running but before it reaches its core functionality.

These differ from **compile-time** or **syntax errors** (which prevent the program from running at all) and from **logic errors** (which appear later during normal operation).



Why Do They Happen?

Environmental Mismatch: The program's assumptions about the system (e.g., files, network, hardware) don't match reality.

Resource Conflicts: Attempts to access locked or unavailable resources early on.

Dependency Issues: Failure to load required libraries or modules.

Configuration Problems: Invalid or missing setup data.



Common Execution Errors at Startup

Error Type	Description	Common Causes	Example (in Python/Java)	How to Fix
Import/Dependency Error	Fails to load required modules or libraries at the start.	Missing packages, version mismatches, or corrupted installs.	Python: <code>ModuleNotFoundError: No module named 'numpy'</code> Java: <code>ClassNotFoundException</code>	Install dependencies (e.g., <code>pip install numpy</code>); use virtual environments; check classpath.
File Not Found	Program can't locate a required file (e.g., config or data).	Wrong path, file deletion, or permission issues.	Python: <code>FileNotFoundError: [Errno 2] No such file or directory</code> Java: <code>FileNotFoundException</code>	Verify file paths; use absolute paths or resource loaders; add existence checks.
Database Connection Failure	Can't connect to DB during init.	Wrong credentials, server down, or network issues.	Python: <code>OperationalError: (2003, "Can't connect to MySQL server")</code> Java: <code>SQLException</code>	Validate creds; add retry logic; use connection pooling.
Out of Memory	Insufficient RAM for loading initial data.	Large datasets or leaks in startup code.	Java: <code>OutOfMemoryError</code> Python: <code>MemoryError</code>	Optimize memory use; increase heap size (e.g., <code>-Xmx</code> in Java); profile with tools like <code>JVisualVM</code> .
Permission Denied	Access to files/directories blocked.	Running without sufficient privileges.	Python: <code>PermissionError: [Errno 13] Permission denied</code> C++: Similar I/O errors	Run as admin; adjust file perms (e.g., <code>chmod</code>); use user-friendly paths.
Configuration Parse Error	Invalid format in config files (e.g., JSON/YAML).	Syntax issues or missing keys.	Python: <code>json.JSONDecodeError</code> Java: <code>ParseException</code>	Validate configs pre-start; use schema validators like <code>JSON Schema</code> .

What is syntax error?

- A syntax error is a type of error in programming that occurs when the code violates the grammatical rules (syntax) of the programming language.
- These errors are detected by the **compiler or interpreter** before the program even starts executing—during the parsing or compilation phase.



What is syntax error?

- **Missing punctuation** (e.g., forgetting a semicolon ; in C++ or a colon : in Python).
- Mismatched brackets, parentheses, or quotes.
- Incorrect keywords or operator usage.
- **Indentation issues** (e.g., in Python, where whitespace matters).



Summary



Test data is data which has been ***specifically identified for use*** in tests, especially in computer program

Data testing is used to verify that a given **set of input** produces some **expected result** to challenge the ability of the program to respond to **unusual, extreme, exceptional, or unexpected input**.

Test data can be done by the tester, a program or function.

The amount of data to be tested is determined by

- ✓ **Time** to produce
- ✓ **Cost** to produce
- ✓ **Quality** of the test data, and efficiency

Data Testing

- After the creation of a system, it is necessary to test data to see if they perform under the **condition** they were created for.
- This includes testing any validation that was created to ensure all data performs as expected.

Types of Test Data

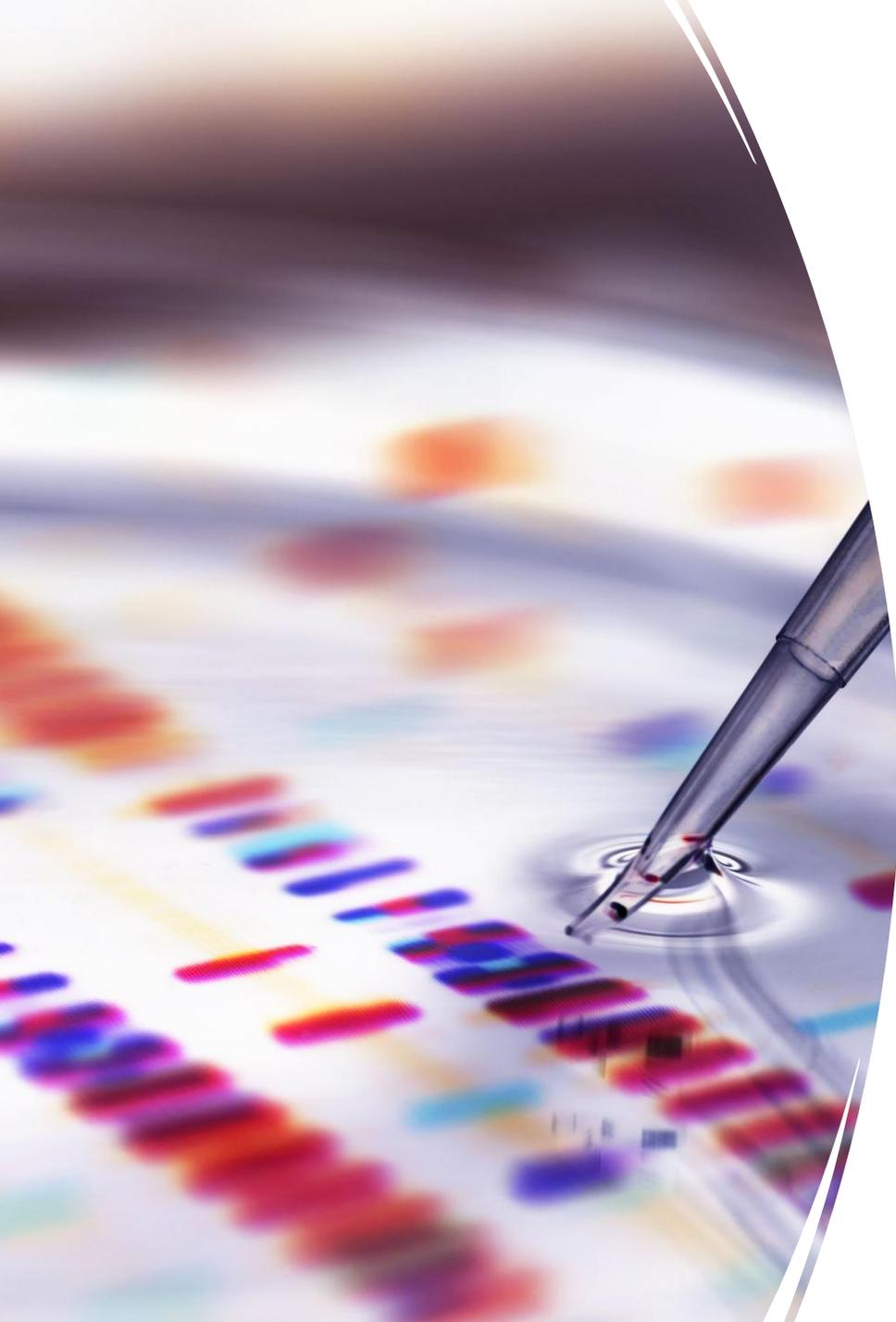
- Normal test data
- Extreme test data
- Boundary test data
- Abnormal/Erroneous test data



Activity

- In 4 groups, research and prepare a 3 slide PPT to demonstrate and explain the concepts of the following **10 min**
 - **Normal** test data
 - **Extreme** test data
 - **Boundary** test data
 - **Abnormal/Erroneous** test data
- Criteria: Definition, give scenario or example to explain the concept and state if the values are accepted or rejected.





Normal Data Testing

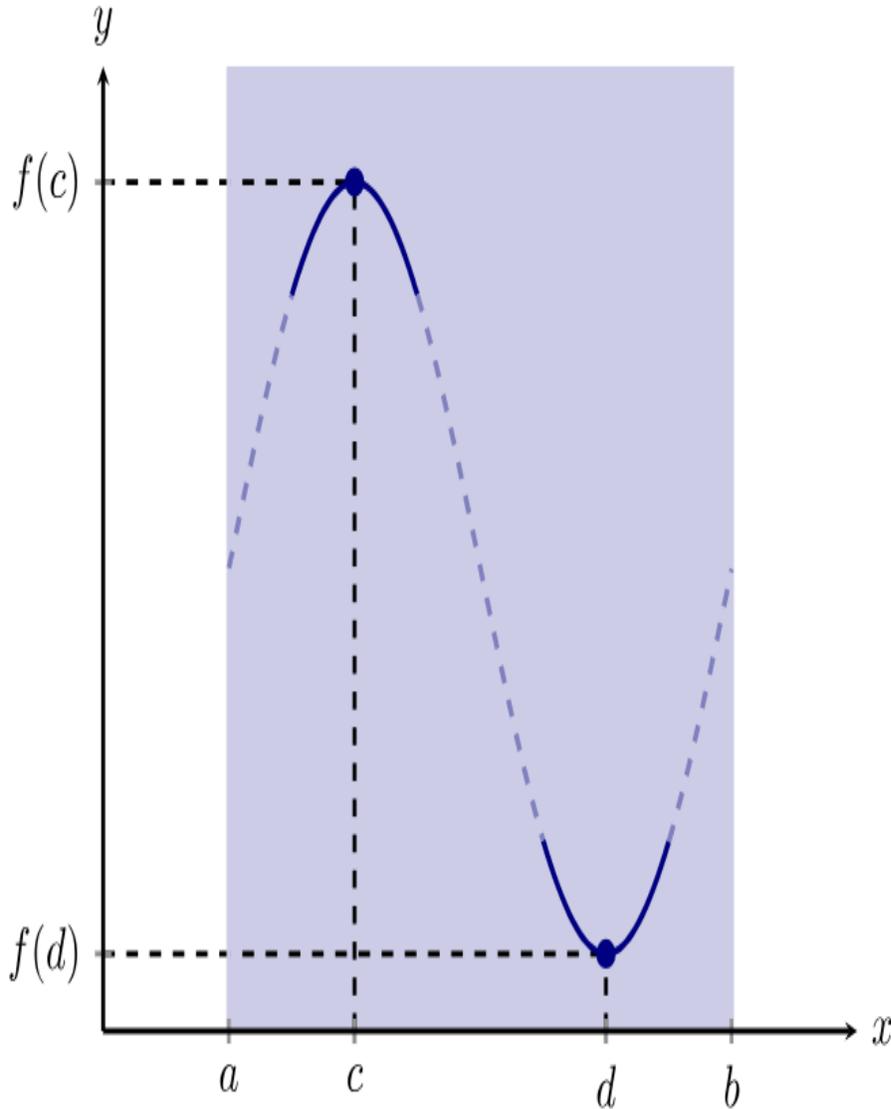
- Normal data is test data that is ***expected*** to be entered into the application) and should be **accepted** by the system and should process it.

Example

- There may be an input form asking for a username. It is *expected* that the user will type in some letters and numbers for this. So normal use data will consist of usernames made of different combinations of letters and numbers.

- Dave321

Extreme test data



- This is testing the very **boundary of acceptable** data is still acceptable and it will be processed in the same way as normal data.
- Extreme/Borderline data is test data at the upper or lower limits of expectations that should be accepted by the system
- Extreme values are used in testing to make sure that all normal values will be accepted and processed correctly.
- For example, the student is asked to enter their assessment mark into the system, the extreme values are **0 and 100**, both values are **accepted** by the system.



Boundaries

Boundary test data

- This is a **pair** forming the upper and lower bound. In a pair, the other value is **accepted** while the other value is **rejected**.
- For example, the user is asked to enter a username with between 1 and 10 characters. The borderline test data would be a username with 1 character and a username with 10 characters.
- Example. If the system accept integer values ranging from 10 to 90, the two pairs of boundary data will be 9,10 where value 9 is rejected and value 10 is accepted and another pair will be 90,91, where value 90 is accepted and value 91 is rejected

Abnormal/Erroneous

- This is data that the program **rejects** as invalid. That might be because it is the wrong data type, contains characters that are not allowed, value falls outside the accepted parameters of the program.

Example, if the user is asked to enter a username with between 1 and 10 characters, invalid test data would be a blank username, or a username with 13 characters, or one that doesn't use standard letters and numbers (emojis or foreign characters, for example).

- If invalid data is presented, then the application needs to handle it properly i.e., not crash. Usually, the user is told that the data provided has been rejected.

Test Plan

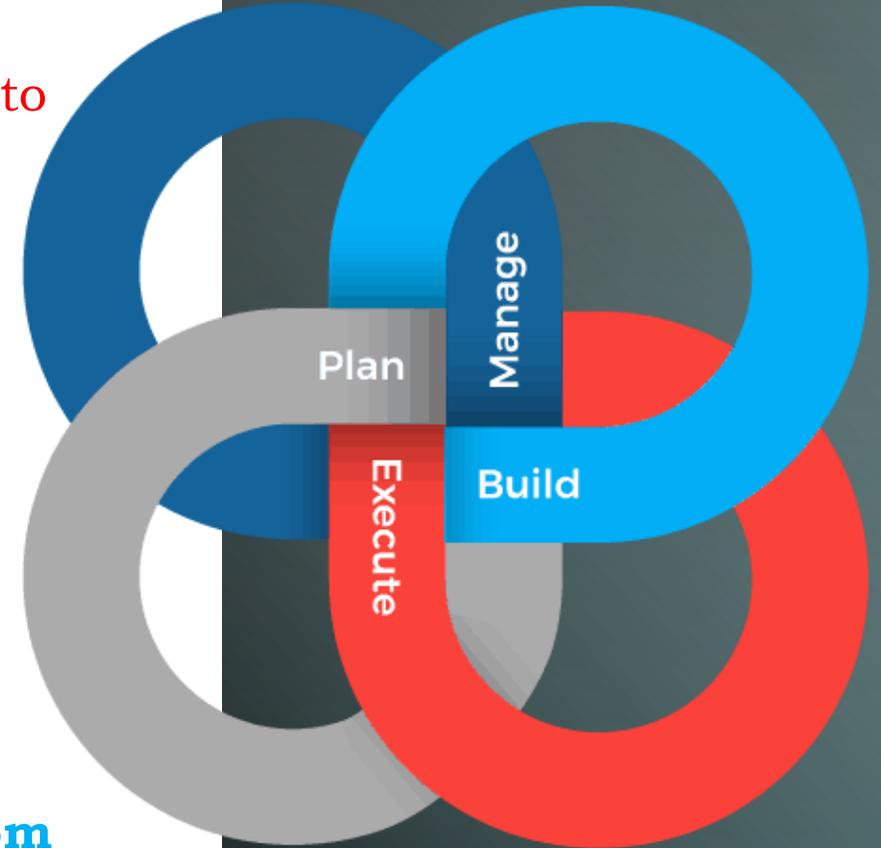
What is a test plan?

A test plan is a formal document which details the tests to be performed on the data/software.

It describes:

- ✓ **The scope of the testing**
- ✓ **The tests to be performed**
- ✓ **The reason for each test**
- ✓ **The data to be used in tests**
- ✓ **The expected outcome of each test**

Once the test has been conducted, **the actual result from the test, along with evidence (e.g. screenshots) is added to the plan.**





TEST PLAN

A test plan should be created for each system module and should list all the different tasks that is going to be performed

A **good test plan** should include

- The list of tests to be performed
- The data to be used in the test
- The type of test –(normal/extreme /abnormal
- The expected outcome of the test

The actual outcome which can be either accepted or rejected

Test Plan

Example of a test plan

The cell should accept age range of 17 - 70. any other data is rejected

Test number	Data entered (test mark)	Type of test data	Expected outcome	Actual outcome
1	20	Normal	Accept	This section would be filled in by the tester and will log what actually happened as a result of each test. If any of the actual outcomes were different to the expected outcome, the module would have to be corrected and then re-tested .
2	45	Normal	Accept	
3	59	Normal	Accept	
4	17	Extreme	Accept	
5	70	Extreme	Accept	
6	Thirty	Abnormal	Reject (error message)	
7	16	Abnormal	Reject (error message)	
8	71	Abnormal	Reject (error message)	
9	-10	Abnormal	Reject (error message)	

Review Questions

Complete the following with a suitable word or phrase from the list below

Validation

extreme

test

normal

erroneous

After the creation of a system it is necessary to _____ data to see if they perform under the condition they were created for. This includes testing any _____ that was created to ensure all data performs as expected. There are three types of data testing that should be done when one creates an information system. A test that is done that is typical or expected and accepted by the system, (_____) another that test data that fall outside of the boundary and is rejected by the system (_____) and one that test the upper and lower limits of expectation (_____).



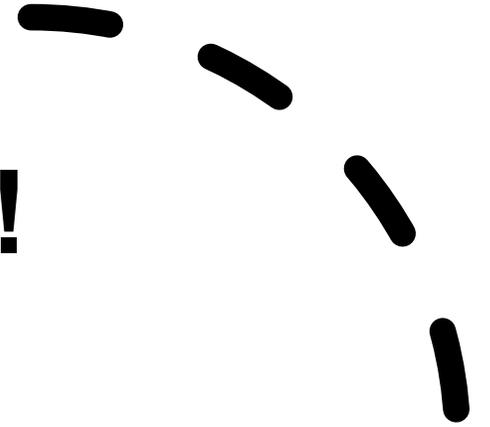
Activity



In pairs, research and prepare for a discussion to demonstrate and explain the concepts of the following 10 min

- Alpha testing
- Beta testing
- Dry run testing
- Blackbox testing
- Whitebox testing

This lesson is Resolved!!!!



Programming Errors



Objectives

At the end of the lesson students should be able to

- ✓ State what is compile error
- ✓ State what is run time errors
- ✓ Describe syntax error in a program
- ✓ Describe logic errors in a program
- ✓ Differentiate between logic errors (Formal and Informal)



Activity- Lesson Review

In peers discuss(Activity) with each other of the following

- ✓ What is data testing
- ✓ Describe Normal test data
- ✓ Describe Extreme test data
- ✓ Describe Boundary test data
- ✓ Describe Abnormal test data



Research Activity

- What are programming errors?



System System

A level of software testing that **evaluates the complete**, integrated software system to verify that it meets the specified **functional(what)** and **non-functional** (specify how well) requirements.

Reasons for System Testing

Holistic Validation

Requirement Compliance.

Defect Detection

Risk Mitigation

Execution Errors

Execution errors at program startup refer to **runtime exceptions or failures** that occur during the initial loading and initialization phase of a software program, right after it begins running but before it reaches its core functionality.

These differ from **compile-time** or **syntax errors** (which prevent the program from running at all) and from **logic errors** (which appear later during normal operation).

Why Do They Happen?

Environmental Mismatch: The program's assumptions about the system (e.g., files, network, hardware) don't match reality.

Resource Conflicts: Attempts to access locked or unavailable resources early on.

Dependency Issues: Failure to load required libraries or modules.

Configuration Problems: Invalid or missing setup data.

Common Execution Errors at Startup

Error Type	Description	Common Causes	Example (in Python/Java)	How to Fix
Import/Dependency Error	Fails to load required modules or libraries at the start.	Missing packages, version mismatches, or corrupted installs.	Python: <code>ModuleNotFoundError: No module named 'numpy'</code> Java: <code>ClassNotFoundException</code>	Install dependencies (e.g., <code>pip install numpy</code>); use virtual environments; check classpath.
File Not Found	Program can't locate a required file (e.g., config or data).	Wrong path, file deletion, or permission issues.	Python: <code>FileNotFoundError: [Errno 2] No such file or directory</code> Java: <code>FileNotFoundException</code>	Verify file paths; use absolute paths or resource loaders; add existence checks.
Database Connection Failure	Can't connect to DB during init.	Wrong credentials, server down, or network issues.	Python: <code>OperationalError: (2003, "Can't connect to MySQL server")</code> Java: <code>SQLException</code>	Validate creds; add retry logic; use connection pooling.
Out of Memory	Insufficient RAM for loading initial data.	Large datasets or leaks in startup code.	Java: <code>OutOfMemoryError</code> Python: <code>MemoryError</code>	Optimize memory use; increase heap size (e.g., <code>-Xmx</code> in Java); profile with tools like <code>JVisualVM</code> .
Permission Denied	Access to files/directories blocked.	Running without sufficient privileges.	Python: <code>PermissionError: [Errno 13] Permission denied</code> C++: Similar I/O errors	Run as admin; adjust file perms (e.g., <code>chmod</code>); use user-friendly paths.
Configuration Parse Error	Invalid format in config files (e.g., JSON/YAML).	Syntax issues or missing keys.	Python: <code>json.JSONDecodeError</code> Java: <code>ParseException</code>	Validate configs pre-start; use schema validators like <code>JSON Schema</code> .

What is syntax error?

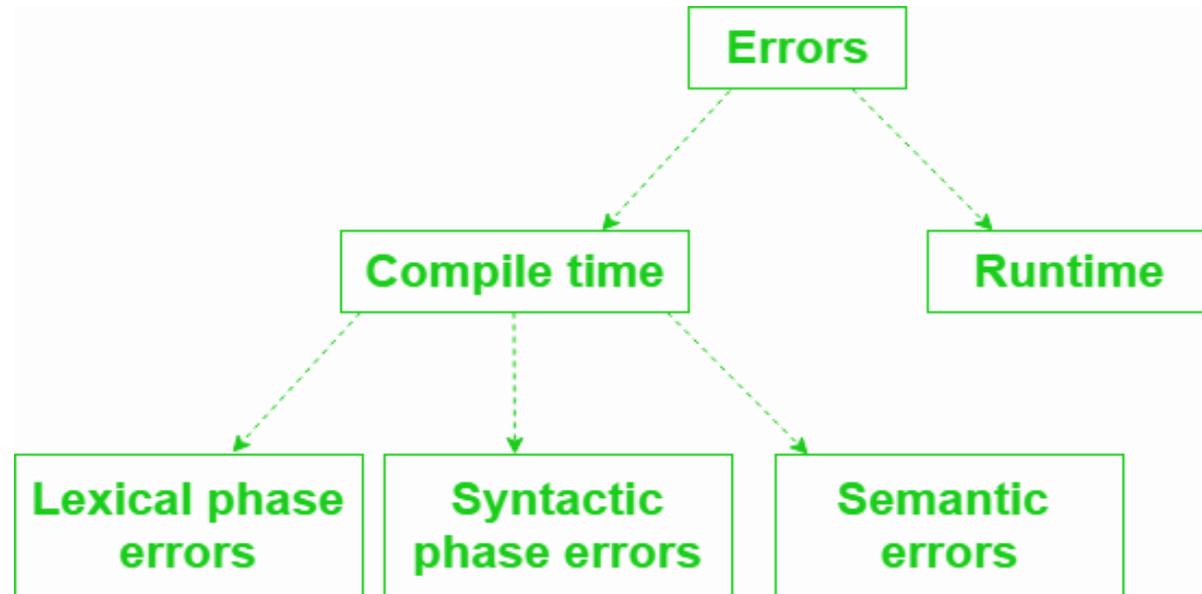
- A syntax error is a type of error in programming that occurs when the code violates the grammatical rules (syntax) of the programming language.
- These errors are detected by the **compiler or interpreter** before the program even starts executing—during the parsing or compilation phase.

What is syntax error?

- **Missing punctuation** (e.g., forgetting a semicolon ; in C++ or a colon : in Python).
- Mismatched brackets, parentheses, or quotes.
- Incorrect keywords or operator usage.
- **Indentation issues** (e.g., in Python, where whitespace matters).

Program errors falls into 2 groups

- ✓ Compile time
- ✓ Run time errors



Compile errors consist of errors that falls in the lexical, semantic and syntactic phases

Run time errors occurs during the code generation phases (backend)

GROUP ACTIVITY

In groups research and make a presentation of not more than 3 slides describing the following errors



- ✓ Syntax Errors
- ✓ Logic Errors
- ✓ Run Time Errors

Assessment criteria's:

- Provides a description of this type of error
- indicates possible causes of occurrence
- Gives an example of the error



Syntax Errors

Syntax errors are errors in the grammar of the program language.

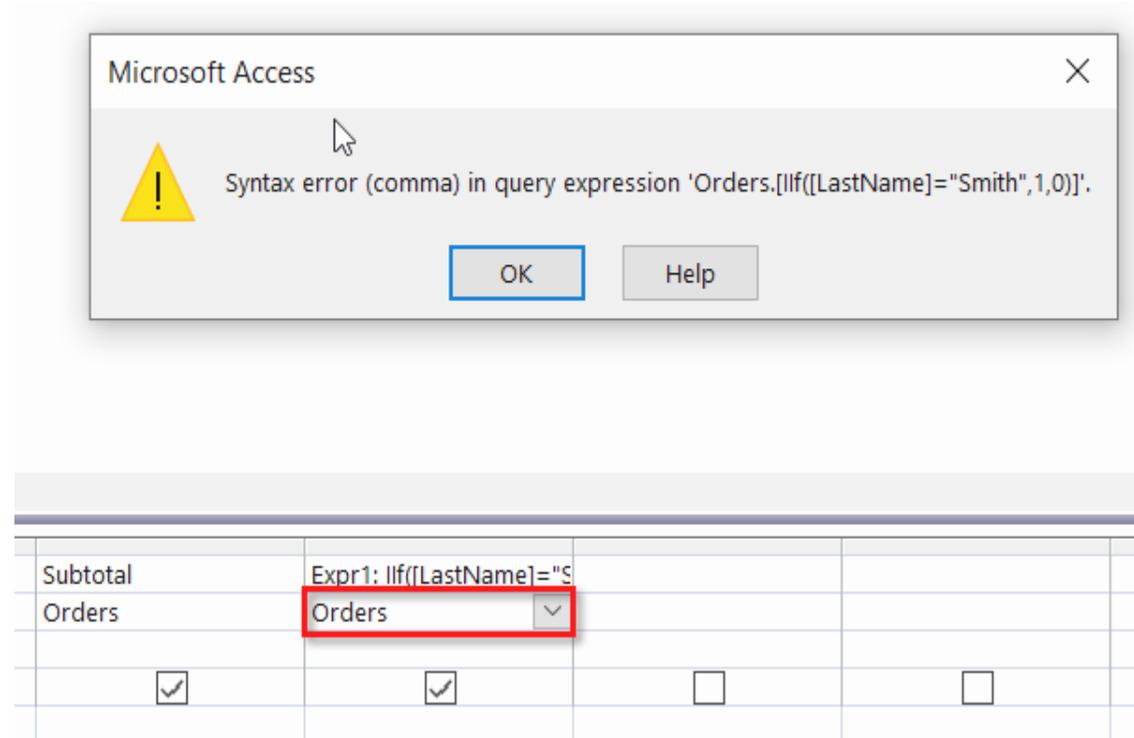
That is, the rules of the language have been broken.

Common errors are **typing errors**, example Selct for Select.

These errors are detected by the **compiler or interpreter** before the program even starts executing—during the parsing or compilation phase.

Other errors are more subtle, for example:

- ✓ an If statement without a matching End If statement
- ✓ a For statement without a matching Next statement
- ✓ an opening parenthesis without a closing parenthesis: (a + b.



Syntax Errors (cont)

Syntax errors are finally picked up during the **syntax analysis stage** of the program's translation.

In the case of an interpreter, the error is reported as soon as it is found.

A compiler reports all errors found in the complete program in its error report.



Logic Errors



A **logic error** occurs when the programmer makes a mistake in their logic for some part of the program.

Example:

If a programmer wants the first 10 positive integers to be output so he writes the following algorithm:

```
FOR Count = 0 TO 10
```

```
OUTPUT Count
```

```
NEXT Count
```

It produces the integers 0 to 10 not 1 to 10.

This type of error can only be found by thorough testing.

This algorithm has no grammatical errors, but it does not produce the correct result

To calculate the average of three numbers (a, b, c), but code uses **average = (a + b) / 3** (forgets c). Input: 10, 20, 30. Output: 10 (wrong; should be 20). The program runs but gives incorrect results.

LOGICAL ERROR

- *A logic error (or logical error) is a mistake in a program's source code that results in incorrect or unexpected behavior. It is a type of runtime error that may simply produce the wrong output or may cause a program to crash while running.*
- *Many different types of programming mistakes can cause logic errors. For example, assigning a value to the wrong variable may cause a series of unexpected program errors*

Example

using the wrong arithmetic symbol,

$a+b$ instead of $a-b$,

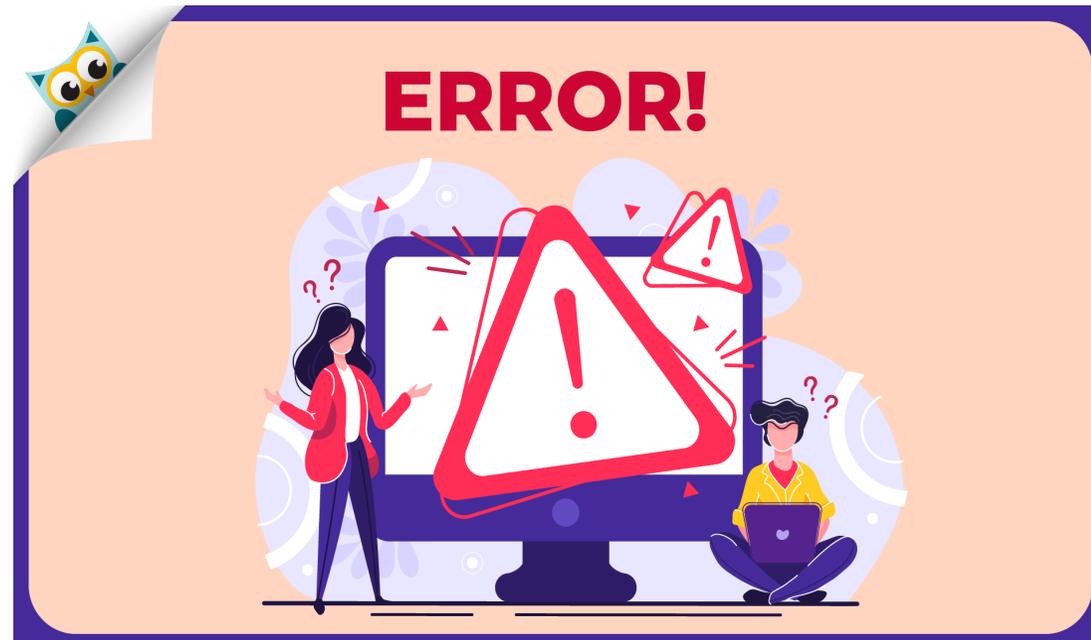
or

to put parentheses in the wrong place,

e.g $(a+b-c)*d$ instead of $(a+b)-c*d$.

These errors should be identified during testing and are all a mistake in the logic of the programmer.

Types of Logic Errors



Formal Logic Error

Formal logic also known as “mathematical logic” is based on the Boolean variable that something can either be true or false.

This is also combined with the logical thinking such as **AND** which says that all ideas are true **OR** only true when any part of the idea is true.

All this along with the **IF then else** statement is the foundation of how computers make choices or reason



Informal Logic Error

Happens when the program runs and follows all the rules of coding, but it doesn't solve the problem the way it was intended.

Used often in debugging a program error

Implementing an algorithm that sorts numbers but unintentionally sorts them in descending order when it should have been ascending.



Run Time Errors

As its name suggests the error is only detectable when the program run

It occurs while a program is running or when you first attempt to start the application.

here are several types of runtime errors.

- ✓ logic error, which produces wrong output.
- ✓ 'memory leak', which causes a program to use up more RAM
- ✓ 'program crash' in which the program unexpectedly quits while running. A program crash is the most noticeable type of runtime error.



Activity- worksheet

Complete the worksheet given and compare your answers with your classmates.

