

1. Difference between primitive and Reference data types
primitive-variable is stored as the value of that variable eg int, float, double.
Reference - Holds an address to information related to that variable.
2. Scope of a variable (hint local and global variable)
Scope of a variable means the lifetime of a variable in the program.
local variables are the ones that are declared inside a function block.
only executes in the function statements.
Global variables are the ones described outside the function.
Accessible by any function.
Method scope - variables declared inside a method. Available in Method
Block scope - Are codes within curly braces {}. Are declared inside blocks
of codes are only accessible by the code between the curly braces which
follows the line in which the variables was declared.
3. Why is initialization of variables required.
initialization - giving a variable a value
Compiler cannot let you use uninitialized variable.
Initialisation is done to avoid logical errors and compilation errors.
Uninitialized variables are not useful to the program.
4. Differentiate between static, local and instance variables.
static variables are variables declared inside a class but outside a
method starting with keyword static. Accessible throughout the class
Local variables - Are variables declared within a method. Limited to the method
starts when a method is called.
instance variables - Are variables declared inside a class but outside a
method. Accessible throughout the class.
5. Differentiate between narrow and widening Casting.
Casting - assigning a value of one primitive data type to another type.
widening - Converting a smaller value to a larger value type. eg int to double
narrow - Converting a larger value type to a smaller value type eg Byte to int

6. Fill in the following table

Type	Size in Bytes	Default	Range
Boolean	1 byte	false	true, false
Char	2 bytes	'\u0000'	'\u0000' to '\uffff'
Byte	1 byte	0	-2^7 to $+2^7 - 1$
Short	2 bytes	0	-2^{15} to $+2^{15} - 1$
Int	4 bytes	0	-2^{31} to $+2^{31} - 1$
Long	8 bytes	0L	$-9.223... \times 10^30$ to $9.223... \times 10^30$
Float	4 bytes	0.0f	3.40282347...
Double	8 bytes	0.0d	-1.8×10^{308} to $+1.8 \times 10^{308}$

7 Importance of Java Packages

— They avoid naming conflicts

— To write a better maintainable code

— Implements data encapsulation

— Makes it easy to search or locate classes and interface

— Uniquely compare the classes in other packages.

8. Controls used when creating GUI applications in Java programs.

- ① Microsoft ActiveX control - Are small program building blocks that can be used to create distributed applications that work over the internet through web browsers.
- ② Swing Control - Swing is a lightweight Java GUI that is used to create window-based applications.
- ③ Standard Widget Toolkit (SWT) - a new class library for creating GUI in Java that allows developers to build efficient, portable applications that directly access the UI facilities of the OS is implemented on.

9. Containers & Components in Java.

Containers - these are the interface between a component and the low level, platform-specific functionality that supports the component. A component is the fundamental user interface object in Java. It is an object having a graphical representation that can be displayed on the screen and that can interact with the user.

10

11 An Event describes the change in state of any object e.g. Pressing a button, clicking a mouse, entering a character in a textbox.

- ① Action Event (pressing a button) It is generated when a button is pressed. Menu-item is selected, list-item is double clicked.
- ② Mouse Event (clicking a mouse) It is generated when mouse is dragged, moved, clicked, pressed or released and also when it enters or exits a component.
- ③ Text Event (entering a character in a textbox) It is generated when value of text area or textfield is changed.

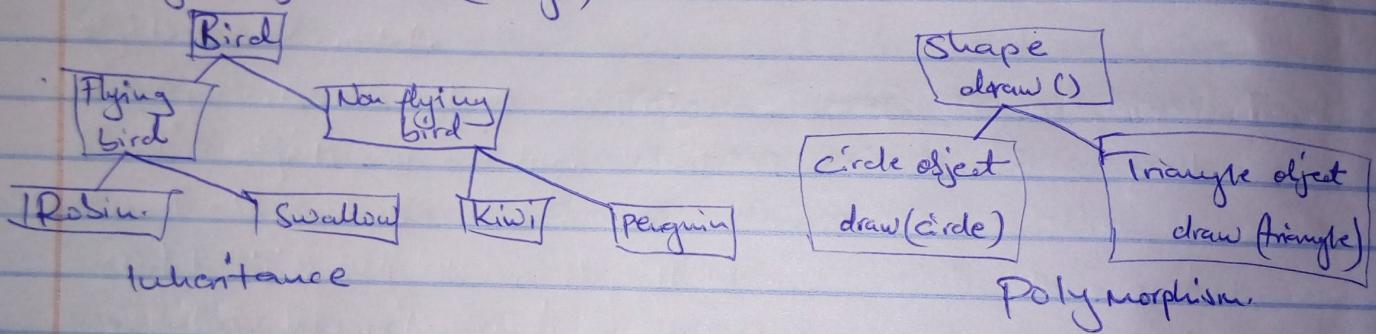
12. Polymorphism and encapsulation - polymorphism (ability to take more than one form) is an OOP concept that allows objects having different internal structures to share same external interface whereas encapsulation is the wrapping up of data and methods into a single unit (class). Only methods wrapped in the class can access it.

Method overloading and Method overriding - Method overloading occurs when two or more methods in the same class have the same name but different parameters whereas Method overriding occurs when the method signature is the same in the super class and the child class.

* Method signature (name and Parameters)

Class and interface - A class is a collection of objects of similar type whereas An interface is a completely abstract class that is used to group related methods with empty bodies.

Inheritance and Polymorphism - Inheritance is the process by which objects of one class acquire the properties of objects of another class. For example bird Robin is part of class Flying Bird which is part of class Bird. Whereas, Polymorphism allows objects having different internal structures to share same external interface. For example Shape draw() circle object (draw(circle)) + triangle object draw(triangle)



① Mutable classes - A mutable class is one that can change its internal state after it is created.

Examples StringBuffer, java.util.Date, StringBuilder.

Immutable

Immutable classes (objects) - the immutable objects are objects whose value cannot be changed after initialization. We cannot change anything once the object is created. Examples Legacy classes, wrapper class, String class.

Only getters (get()) method are available for immutable objects.

Suitable when making changes in-place without allocating a new object.