## OOAD

## Java



### Compilation Process

javac

→ class file → loaded into Classloader → check for bytecode → interpreter

→ execute

→ send it to the

CPU/hardware

class file = bytecode

### terminologies

JVM

JDK

software kit

JRE

Java Runtime Environment

JVM

### program I/O

System.out.println()

Scanner(System.in )

nextInt()

nextLong()

…

nextLine() string

## IDEs

IntelliJ

VScode

Eclipse

Visual Studio

### Online env:

replit

### Java as a language

Statically typed Language

Strongly Typed Language

### keywords

boolean byte char

false true

int short long

float double

## Data types

### Primitive Data Types

int 4

short 2

long 8

float 4

double 8

char 2

boolean 1 bit

byte 1 byte

float

12.36

1236 x 10-2

### Non Primitive Data Types

Reference Types

Arrays

Strings

Class

Objects

Interfaces

## Operators

### arithmetic

+

-

\*

/

% modulo

### assignment

=

compound

+= a += b

a = a + b

-= a = a -b

…..

### relational

<

>

<=

>=

==

!=

### logical

&& and

|| or

! not

### unary

-

a = -100

++

post

pre

### bitwise

& and

| or

^ ex-or

### shift

<<

>>

>>>

<<

0011 1100 x

0111 1000 y

0001 1110 z

0011 1100 x

1111 0000 x << 2

0000 1111

+1

0001 0000

1010 0000

1010 0000 0000 0001

0001 1111 31

0000 0011 3

0000 0100 -4

### terniray

## Flow control

### if else

if

else if

else

### switch

switch

case

break

default

### while

while

do while

break

continue

### for

for

break

continue

### for-each

loops automatically

assigns too

features:

no index

can not be effectively used to change contents of the array

moves only forward

moves only in single steps

## Arrays

* dynamically allocated
* continuous memory allocation
* objects
* [ ]

int [ ] arr = {10, 20, 30}; //length is 3

arr.length → data

String stra = “atlas” // length is 5

stra = “amazon” // length is 6

stra.length() → function

## Strings

### String

* objects
* immutable

String literal

* String Constant pool
* JVM optimize

new operator

* dynamically allocated
* heap

### String methods

str.length()

str.toUpperCase()

str.toLowerCase()

str.indexOf(“”) the index of that particular sub-string

str.charAt() the character the specified index

str.isEmpty()

### StringBuffer

a t l a s t

01 23 4

### StringBuilder

faster than StringBuffer

not thread-safe

## OOPs

### general concepts

#### vocabulary

state

(values)

properties

behaviour

methods

identity

class (memory) division

objw objx objy



#### class declarations, in general:

modifiers

public

private

constructor

methods

#### declaring member variables

various kinds:

within a class → fields

within a method (or block of code) → local data

in method declaration → parameters

#### method name (conventions)

speak

speakLoudly

getData

runFast

changeRoomTemperature

#### objects

Bank objy = new Bank(234682, 3000, "savings");

Bank objy → declaration

new → instantiate

Bank(xxx, yy) → initialisation

#### garbage collection

JRE deletes objects when it feels that those objects are no longer being used

runs automatically

an object can be deleted, when:

-- no references to that object

#### Features

Encapsulation

Inheritance

Polymorphism

function/methods overloading

C++ Java

operator

C++ Python

Abstraction

## Polymorphism

4 + 6

“hello” + varx

Static

compile time

go\_for\_dinner(Saturday)

Runtime

Run time

Function Overloading

* num of parameters
* data types of parameters
* order of parameters

## Encapsulation

capsule

hide the data

## Object & Class Relationships

Association

Composition

Aggregation

### Association



association manages:

one-to-one

one-to-many

many-to-many

two forms of association:

composition

aggregation