Friday, April 22, 2022 9:34 AM

We have done
Package ,import , scopes , access specifiers , primitive data types, wrapper classes, String, StringBuffer , StringBuilder

CHARCTER ENCODING ----------

1. Unicode
2. ASCII

A ==========> binary representation

1)

Use ASCII encoding ---- 1 byte for each character
0100 0001

How many combinations of 0 and 1 in 8 bits ?
2 raised to 8 = 256

So we can represent 256 chars using ASCII

2) Use Unicode encoding --------- 2 bytes for each character
0000 0000 0100 0001
How many combinations of 0 and 1 in 16 bits ?
2 raised to 16 = 65536

So we can represent 65536 chars using UNICODE

OOPS concepts in Java ----------

Object Oriented Programming System

What is a class ? Class is a design of the object . It includes properties/data members/attributes/fields and behaviors methods/functions

What is an object ? Object is an instance of the class. It is a real entity based on the design provided by the class.

Architect plan = class
Building = object

---

4 Major Pillars of OOP

1. Abstraction
2. Encapsulation
3. Inheritance
4. Polymorphism

3 Minor Pillars of OOP

1. Strong Typing
2. Persistence
3. Concurrency

1. Abstraction = Selection of **Relevant** Properties and Behavior in a class

   Person ---- height weight gender color name age dob bloodgroup education degree, spouse, sibling, mother, father, address, adhaar card number, nationality, income, mobile number, email number, grandmother greandfather, relationship status, insta, fb, twitter, religion ,race ,occupation, language, mother tongue, hair color eye color, sugar, heamoglobin , heart rate ……

   Tree----------

2. Encapsulation =

   One class **binds together** the properties and behavior
   The properties are **hidden** and allowed restricted access

   Setters ? Conditional SET -- set value only if appropriate = restricted access

---

Relationships between classes

HAS A
IS A

HAS -A ====== One class HAS-A reference of another class as a property .

```
class A   Student
{
        int x;
        B  obj;    MyDate   dob
}


 class B   MyDate
   {
   }
```

Ex = Write a class

study.hasa.TechnicalBook

Properties

bookName , cost
MyDate dateOfPublication
String[] authors

MyDate

Day , month ,year

2 constructors ---- no args constructor { keep it blank } , parameterized
getters and setters

User
Main
create a TechnicalBook using default constructor
Show the year of publication

Create another TechnicalBook using parameterised constructor
Show the total number of authors

Show the total cost of both the books

HW 1--

write a class study.hasa.hw.Product

Product
name
desc
unitCost
MyDate expiryDate //USE the MyDate that we wrote in class

2 constructors, getters and setters
User
Main
create 10 products add them to array , take values as hardcoded !!

Show a menu
1. Modify a product -- ask which product index to modify , ask which property to modify
2. Show all products , that are not null
3. show total cost of all products in the array
4. show names of all products
5. remove a product ------ set the array[i] = null
6. quit

Quitting a LOOP ----------

1. Use break
2. Use System.exit(0)
3. Use flag = false
4. Return

String class
split()

HW 2 ---

create a class study.has.User3
Copy the code of User2 in User3
Then replace the MyDate and authors creation code using split as discussed in class

---

Relation = ISA relation = INHERITANCE relation

Parent - Child relation ??

Child **isa** type of Parent

Child **inherits** attributes and behavior from Parent !!! But Child can also have its own attributes and behavior !!

Base -Derived relation !!!
Base = Chair --- it should be elevated from ground , it should have seat rest, back rest
Derived Chair = Rotating Chair isa Chair
Rotating Chair = from base --- elevated ground , seat rest, back rest
its own ---- rotate wheels , hand rest, movable back rest, height adjuster

Java Naming Convention----------

Parent	Base	**Super**
Child	Derived	**Sub**

SUPER- SUB relation is called as IS A relation , Inheritance !!!

Inheritance creates a Hierarchy !!!

Syntax for establishing IS-A relationship between classes --------

Keyword --- extends !!

Create ISA relation between
Class Person = Super
Class Employee = Sub

Ex1 --- study.isa.Person
Employee

UserOfHierarchy
main

HW 1 and 2 are given ABOVE in notes.

HW 3 Write a class study.hw.Message with 2 constructors getters and setters

Properties are

String senderName, receiverName, message

Address receiverAddress

Write a class study.hw.Address with 2 constructors getters and setters

Properties are

area, city, state, country, pin

In the same package create User

Main

Create array of 5 messages, in hardcoded way

Ask the user to enter a city name

Show details of all messages having receiver in the entered city

HW4 Write a class study.hw.SumItUp having main that will accept few numbers from user in a coma separated string

Use split method

Show the sum of all numbers entered

HW5 Write a class study.hw.Looping having main

ask the user to enter a balance value;

show menu

1. Deposit --- accept a value and add to balance
2. Withdraw ---accept a value and deduct from balance

Minimum balance is 100

If balance becomes less that 100 then do not deduct , just print message "not enough balance"

3. Show balance
4. Quit

HW6 Write a class study.hw.StudentNames

Properties ---- String[] names (firstname lastname separated by space)

2 constructors

Methods ---

Public void addName(String name)

Public String getNameOnIndex(int index)

Public void showAllNames()

Write a study.hw.UserOfNames

Main

Menu

1. Show all names
2. Add name

3. Get name on index
4. quit