Object Oriented Programming

ফাংশন গুলো অবজেক্ট এর আচরণ প্রকাশ করে। একটি ফাংশন একটি অবজেক্ট এর সাথে অন্তভুক্ত হয় তখন তাকে মেথড বলে।

Inheritance:-

অবজেক্ট গুলু অন্যান্য অবজেক্ট এর সাথে সম্পর্ক মেনে চলে। এই সম্পর্ক দুই ধরনের হয় IS-A এবং HAS-A.

IS-A : এই সম্পর্কটি দেখাব তখন তাকে বলে Inheritance.

HAS-A : এই সম্পর্কটি দেখাব তখন তাকে বলে Object Composition.

ইনহেরিটেন্স মূলত একটি Tree Relationship.

মানে একটি অবজেক্ট থেকে ইনহেরিট করে আসে আর যখন আমরা অনেকগুলো অবজেক্ট নিয়ে একটি অবজেক্ট তৈরি করব তখন সেই নুতুন অবজেক্ট টি হল মেইড-আপ (Made-Up) বা নতুন তৈরি করা অবজেক্ট। এই ঘটনা টি হল কম্পোজিশন।

Inheritance এ Child Class, Parent Class এর সবগুলো Field এবং method গুলো Inherit করলেও এর Constructor কে করেনা। এক্ষেত্রে Child Class এর Constructor Call করতে পারে।

Rules of Overring method:-

1. সাব ক্লাস এর মেথডের রিটার্ন টাইপ সুপের ক্লাস এর মেথডের রিটার্ন টাইপ একই বা এর যে কোন সাব ক্লাস হতে হবে।

যেমন,

Class Shape{ }

Class Circle extends Shape { }

Class ShapeFactory

{

Public Shape newShape() { }

}

Class CircleFactory extends ShapeFactory

{

@Override

Public Circle newShape() { }

}

একে বলে Covariant Return Type.

1. যেসব মেথড ইনহেরিট হয় না তাদের ওভাররাইড করা যায় না।

Ex: private, static, final

Super Keyword:-

Super Keyword ব্যবহার করে super Class এর Private Field গুলো Access করা যায় না।

Polymorphish:-

Polymorphism এর ক্ষেত্রে যে কোন অবজেক্ট রেফারেন্স জানার জন্য getClass() কল করে এর ক্লাস অবজেক্ট টি পেতে পারি। এবং এর getTypeName() কল করে কোন অবজেক্ট টি আসছে তা জানা যায়।

Ex:-

List obj=new ArrayList();

System.out.println(“Type name:”+obj.getClass().getTypeName());

String superType=obj.getClass().getSuperClass().getName();

UPcasting & DownCasting:-

Upcasting:-

সাব ক্লাস এর অবজেক্ট কে সুপার ক্লাস এর রেফারেন্স এ অস্যাইন করা কে বলা হয় Upcasting .

Ex:-

Liquaid lq=new Coffee();

DownCasting:-

Coffee cp=(Coffee) lq;

একে বলা হয় Downcasting.

অথবা,

( (Coffee) lq).getLiquidName();

Downcasting করতে হলে আমাদের নিশ্চিত করে জানতে হবে যে টাইপ এ কাস্ট করা হচ্ছে তার টাইপ এবং অবজেক্ট টাইপ একই।

Instanceof keyword এর মাধ্যমে নিশ্চিত হওয়া যায়।

If( lq instanceof Coffee)

{

Coffee cp=(Coffee) lq;

}

Abstract Class:-

Abstract Class এ Field বা প্রপাটি, মেথড এবং কন্সট্রাক্টর থাকতে পারে, তবে একে ইন্সট্যান্সিয়েট করা যায় না। এর সাব ক্লাস গুলোকে বলা হয় Concrete Class.

Rules:

* Abstract Class ক্লাস এ কন্সট্রাক্টর থাকতে পারে। তবে যেহেতু কোন ক্লাস এর অবজেক্ট তৈরি করার সময় এর কন্সট্রাক্টর কল হয়, সুতরাং এর কন্সট্রাক্টর কে সাব ক্লাস এর কন্সট্রাক্টর এর Super Keyword মাধ্যমে কল করতে পারে।
* Abstract Class কে Method বা কে কখনও Final Declare করা যায় না।
* Abstract Class এর Constructor Private হতে পারে না।
* কোন রেগুলার ক্লাস এর সাব ক্লাস Abstract হতে পারে না।

Pure Abstract Class:-

যে এর সবগুলো method abstract তাকে Pure Abstract Class বলে।

Interface:

Interface এর সবগুলো Method public abstract এবং Variable গুলো public static.

Encapsulation:-

Object Oriented Programming এর আরেকটি খুবই গুরুত্বপূর্ন বৈশিষ্ট, ক্লাস এর মধ্য Data গুলো লুকিয়ে রাখা যায় এবং সুধু মাত্র ঐ ক্লাস এর মেথড গুলর মাধমে Access করতে দেওয়া, একে বলে Encapsulation.

Rules:-

Override করার ক্ষেত্রে মডিফায়ার গুলো সাধারনত একই হয়। তবে সুপের ক্লাস এর মেথড যদি প্রোটেক্টেড হয়, সাব ক্লাস এ সেটিকে পাবলিক করা যায়, তবে উল্টোটা সম্ভব নয় ।