

Pure Virtual Fun & Abstract Class

(Unit - 4)

Q) What is a PVF?

A. Normally, it is not mandatory for der class to o/r a VF..

- However, if a VF is declared as PVF, then it becomes mandatory for the derived class to o/r this VF.
- If der class doesn't override then an error will be generated by compiler. The compiler won't allow the program to run.

Q) Why is a PVF needed?

A. The purpose of a PVF is to force the der class to o/r a base class VF

- Sometimes, a programmer may forget to o/r a VF in der class.

- For eg :- In the eg. of employee class, the HR class doesn't o/r the raise_sal() fun.
- Now, HR class didn't o/r the raise_sal() fun, so when HR-obj calls this fun, then base fun will be called. This shud not be allowed.
- To avoid this situation, raise_sal() fun is declared as PVF in base. Now, HR class must o/r this fun o/w compiler will not run the program.

Q) How is a VF declared as pure?

A By using a "= 0" after fun name.

eg Virtual void fun() = 0
// Pure VF //

Note - Fun must be virtual.

• Normal fun can't be declared as pure

void fun() = 0 // wrong.
// Non vir fun can't be pure.

Abstract Class

Q) What is an abstract class (AC)?

A. Any base class with at least one PVF is called an AC.

- It may contain ^{other} normal var & fun but there must be at least 1 PVF.
- A PVF is not defined in the base class.

Q) What is the purpose of an AC?

A. An AC is a blank class. It doesn't define the PVF.

- It simply works a framework for the der classes.
- So, der classes must implement all the PVF in base class.

Q) Characteristics of an AC?

A • obj of AC can't be created
bcoz it has undefined PVF.

• However, ptr & ref of AC can be created.

• An AC is always used as base class.

• All der classes must override the PVF in AC.

• Purpose of AC is to provide a framework to der classes & to provide base ptr/ref to achieve RTP.

• AC can have normal var & fun other than PVF.

Q) Char of a PVF?

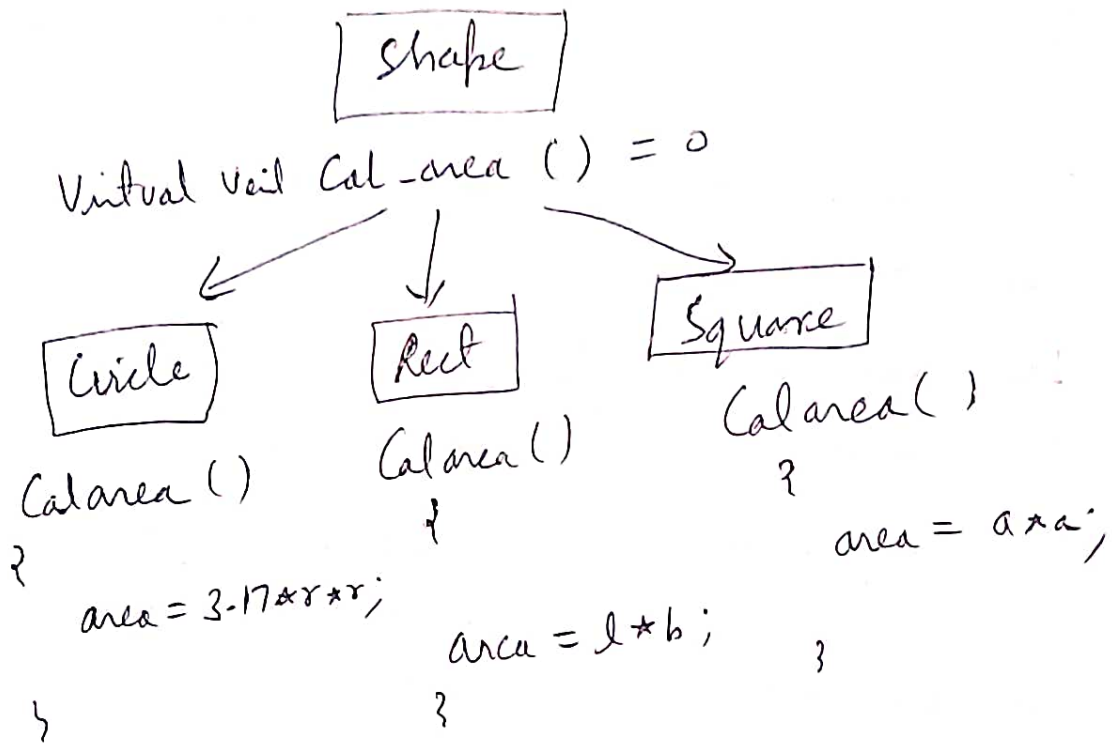
A • A PVF is declared using = 0

• A PVF is not defined in base class.

• Der class must define a PVF.

• A class w/ PVF can't create obj. & it is called an AC.

Example of a PVF & AC



- Shape is an ~~ab~~ AC. It contains a PVF Cal-area()
- This class is inherited by circle, rect, sq.
- These der classes must define the Cal-area() fun.
- If any class doesn't define the fun then it becomes an abstract class well.
- // see code //

Virtual fun

PVF

1) VF is defined in
base class

2) base containing VF
doesn't become AC

3) base class obj can
be created

4) A VF may or may
not be redefined
in der class.

1) Not defined in base

2) base containing PVF
become AC

3) base class obj can't
be created

4) A PVF must be
redefined by der
class.

~~5)~~