

OOPs Assignment

Inheritance using Abstract class,

1. Create an abstract class **Machine** with an implemented / concrete method **rotate** and an abstract method **crush**. Create a class **Juicer** which extends this abstract class **Machine** and implements method **crush**. Now, add another method **filter** in the class **Juicer**. In main method,
 - a. Create an object of type **Juicer** and calls its **crush**, **rotate** and **filter** methods.
 - b. Create an object of type **Juicer** with reference variable of **Machine** (**Machine m = new Juicer**). Check the methods available to **m**.
2. Above we created an abstract class **Machine** with an abstract method **crush** and a concrete / implemented method **rotate**. Now create another abstract class **Mixer** which extends the abstract class **Machine**, has method **crush** implemented and has an additional concrete method **blend**. In main create an object of class of **Mixer**. Call methods **rotate**, **blend** and **crush**.
3. Create 2 abstract classes **Abs1** and **Abs2** each with different implemented methods **doAbs1()** and **doAbs2()** respectively. Check if you can create a class **Temp** which extends both the abstract classes. (Note- A class cannot extend 2 classes simultaneously)
4. Check –
 - a. Can we create an object of an abstract class?
 - b. Can we declare a class abstract with no methods in it?
 - c. Can we declare a class abstract even if it does not have any abstract method?
 - d. Can we declare an abstract class which has both abstract as well as implemented methods?
 - e. Can a class extend 2 or more abstract classes?
 - f. Can an abstract class extend 2 or more abstract classes?
 - g. Can an interface extend 1 or many abstract classes?
 - h. Can an abstract class implements 1 or many interface?
5. WJJP2 create an abstract class **Parent**. Add an abstract method **cook()** in it which has only definition and one method **wash()** which also has implementation. Create class **Child** which extends **Parent** and add the missing method implementation. In main, use both the methods by creating instance of the concrete class.
6. Show use of parameterized constructor of parent class (**Child** class invokes parent's parameterized constructor) using **super**