	enum can become member of .java file similar to class and interface. For enum .class file is generating while compiling. enums are mainly for grouping similar kind of fixed no. of constants. All constants should be unique, every constant should be a valid java identifier. All constants in an enum should be separated by a comma. By default, every enum constant is a public, static and final. Inside an enum, not only enum constants, we can also incorporate attributes, constructors, IIBs and SIBs also.
1.	**Which keyword is used to define an enumeration in a java file?** enum
2.	**How can one access enum constants?** To access an enum constant use enumName.enumConstant
3.	**What happens when one tries to access an enum constant which is not present in enum?** When one tries to access an enum constant which is not present in an enum definition then it will result in compile time error.
4.	**Can an enum be defined inside class body?** Yes, definitely.
5.	**Is it necessary to end list of constants of an enum with semicolon (;)?** It is optional if enum contains constants only. But if there are other members for enum then constants list should end with semi-colon.
6.	**Specify some methods related to enum with purpose.** <u>values()</u> It returns every constant of an enum in an array. <u>ordinal()</u> To read an index of constant it is useful. It returns index of constant. <u>valueOf(String enumConstant)</u> If enumConstant is a String which is representing a constant then it returns the same string. If specified constant is not available in an enum then it will result in runtime exception.
7.	**Why while using valueOf(String enumConstant), it doesn't give compile time error if specified enumConstant is not defined?** While using valueOf() method if specified value is not available you will be getting runtime exception. As compiler can't identify what is the content under double quotes so it is unable to identify whether constant is defined or not. Thus while running if is it unable to find specified constant it will give runtime exception.
8.	**How to use enum defined in one class into another class?** className.enumName
9.	**For every enum there should be minimum <u>1</u> constructor. If you are not providing a constructor compiler keeps** <u>default no arg constructor</u>.
10.	**When does enum constructor gets executed?** Every enum constant is a static. To use any member of enum, enum should be loaded to memory. While loading a constant to memory constructor will be executed. Corresponding constructors are for every enum constants are executing. Once complete enum gets loaded then only you can use any constant.
11.	**What happens if all enum constants are int argument and no int argument constructor is defined?**

	If programmer has not defined int argument constructor then compiler will keep no arg default constructor which will result in compile time error as for constants corresponding constructor is not available.
12.	**If an enum contains a non-static member, non-static method and 4 constants. How many copies of non-static member and non-static method will be created?** For each constant a separate copy of non-static member as well as non-static method will be created. So 4 copies.
13.	**If you want to modify a constant specific method behavior then what should be done?** If you want to modify a constant specific method behavior then use constant specific class body.

14.	**How can one specify constant specific class body?** inside enum after the constant for which you want to specify
	enumConstant
	{
	    //method implementation
	}

15.	**In case of supplying enum to switch what should be case labels?** While supplying an enum to switch case labels should be constants of an enum
16.	**Why arguments are required for enum constants?** For similar type of constants, to maintain varying value of common property of different constants.
	Prototype Design Pattern Almost like a duplicate object,whatever data is there using that data, create a new object. It is creational design pattern. It is always advisable to provide implementation of Prototype Design Pattern as it facilitates to develop new object using content of existing object. It is cloning of an existing object instead of creating new one.
	Builder Design Pattern Building complex objects with various ingredients : 1st object containing 3 ingredients of different types, 2nd object containing 5 ingredients of different types, 3rd object containing 4 ingredients of different types and so on. Builder class is based on ingredients. Every Builder object must have container which can accommodate variety of ingredients and in varying quantity.
	Object Pool Design Pattern Advisable to use object for some time after it has been created. Whatever object you have created, you should use it and save it at safest place, again re-use it and save it at safest place. Pool means something like multiple elements. Object Pool should have mechanism to store an object for the stipulated amount of time inside a pool. It is using HashTable: It is a container, we can store elements with a key. In ObjectPool abstract class there are 2 HashTable type containers : locked, unlocked are used. Every element is having a unique key. To read element back using unique key it is possible. If at all the object is being used by one of the user then another user shouldn't be allowed

	to use the object.Once object is being used by one of the user then it should be added to locked container and it can't be used by any other user until it is freed up. Once object usage is over it should be added to unlocked container object in unlocked container can be used by any of the user. In the constructor, choosing expirationTime, initializing 2 containers. 3 abstract methods : create(), expire(), and validate().As its implementation is dependent on type of objects so not implementing in the ObjectPool class. Implementation varies from one type of object to other type. in checkout() → check whether there is any 1 element which isn't expired and validated one by one checking from unlocked container is there 1 element which is not expired and validated if it is then allot it to user and add to locked container. in checkIn() →once object is used it is removed from locked container and added to unlocked container with current time.

	Why do we require arguments to enum constants? Assume there is a common property to multiple constants and its value is varying. This varying value can be supplied as an argument to the constants. In order to supply varying value of common property of multiple constants.
1.	**What is Enumset and why do we require?** Enumset is a type of set, which allows only unique elements. Enumset is used to group constants of one enum. You can consider any number of constants into one Enumset from one enum. You can consider all constants (or) few constants, but from different enums into one Enumset.
2.	**Is it possible to develop methods without constants inside enum?** By default NO. But there is another way you can end one empty statement with semicolon, then you can develop any number of methods.
3.	**What are the allowed access levels to enum constructor?** Only private (or) default. The reason is all enum constants are declaring within enum itself not outside.
4.	**If enum containing a main method, then is it possible to run that enum or not?** YES
5.	**Is it possible to extend a class to enum?** No. While compiling compiler making every enum as a subclass to java.lang.Enum.
6.	**Is it possible to implement interface to enum?** Yes, to provide any contracts to enum.
7.	**Is it possible to declare enum as an abstract?** NO. enums are not involving in the inheritance.