UNIT-II
Classes & Objects

Creating Classes, declaring objects:

To work with OOP languages, one should have knowledge of two things.
1. Class
2. Object

Class is an ADT (Abstract Data Type). Simply, it is a user defined data type. It is used to group the variables and functions that are required for an object. It is like a template of object or a blue print of object. We use the keyword "class" to create a class.

Syntax :
class <Class Name>
{
Data members
Member functions / methods
}

Object is a named instance of class. It is like a variable of a primitive data type. When we create an object to a class all the features of the class are abstracted to the object. We can make use of the features declared in class through object with a "." DOT operator called PERIOD operator. A method is a function defined with in a class that does some process. We can also pass parameters to these methods as we did in C language.

Java is dynamic, i.e. each object in java must be allocated with memory with the keyword "new". When we declare a reference for class, it cannot be used directly without instantiation (created with instance i.e. allocated with memory). When the reference is instantiated, then it becomes object.

A language to be an OOP, should have some standards and are called oops concepts. They are:

Abstraction : Abstracting features of class into object in different ways.

Encapsulation: Selective hiding of class members in object

Polymorphism : One thing in many forms

Inheritance: Getting features of a class into another.

Different benefits of oops achieved by these concepts are better memory management, high data security, easy identification of methods that process data, reusability of code etc.

Q. Write about class and objects in java.

---

static fields and methods:

static : is a keyword used to declare class members. The class members are allocated with memory soon after program execution starts even if object is not created. So the static members can be called with class name without object. The Method header

```
public static void main(String args[]){
process;
}
```

static variables also get allocated with memory once. When a number of objects are created for the class, still the static variables will have memory allocated once and shared among all the objects. Static variables can be declared only with in class. They cannot be declared with in a method.

In java, methods can also be declared as static. If a method is declared as static, it can be called with class name without object. A static method of a class can also be called directly with in another static method of same class.

Static Block : a class can have a static block for initializing static variables of the same class.

Ex:

```
class MyStaticTest{
static int a;
static{
a=10;
}
}
```

Q. write about static features of java

Constructors:

Constructors: constructor is a special method of the class that has same name of the class name and will be invoked automatically whenever the class is instantiated, i.e. allocated with memory with new operator. The constructor can also take arguments. The class may have any number of constructors where each has different number of arguments or different data type argument implementing functional polymorphism. The constructor that has no arguments is called default constructor.

Each class in java must have a constructor. If no constructor is written in a class, then the JRE declares a default constructor in that class with empty definition. If a constructor is declared in a class with arguments and no default constructor written, then the JRE won't create a default constructor for that class. As these methods (constructors) are used to construct memory for the member references declared in a class, these are called constructors. And these can also be used for initializing the values of data members. Though the constructors are methods, but not used for general process like normal methods, the constructors cannot return any value and these have no return type.

Q. Write about constructors

‘this’ keyword:

The functions in C are called methods in java, as they are declared within a class. A method is piece of code identified with a name. A method has two aspects, Method definition and method call. A method is defined with in a class and can be called with object. The method name includes access specifier, access modifier, return type, method name and arguments.

The first line that includes method name, arguments etc is called function signature in C and the same is called method header in java. The method body includes the process to be done. A method once defined, can be called with object any number of times. When a number of objects are created for a class, the methods are allocated with memory once and shared by all objects, and it is called functional abstraction, thus methods provide better memory management.

When a method is called by an object, the object puts itself into that method definition and gets processed. And the object that puts itself into method to undergo process is called “current object”. This current object is referred with keyword “this” within that method.

The “this” is a keyword used to refer current object and also we can use “this()” to pass arguments from one constructor to another constructor of the same class, if the class has more than one constructor.

Q. Explain this keyword

<u>OOPS Concepts</u>

Java is dynamic, i.e. each object in java must be allocated with memory with the keyword “new”. When we declare a reference for class, it cannot be used directly without instantiation (created with instance i.e. allocated with memory). When the reference is instantiated, then it becomes object.

A language to be an OOP, should have some standards and are called oops concepts. They are:

Abstraction : Abstracting features of class into object in different ways. It is further classified into data abstraction and functional abstraction. The functional abstraction is done once and shared by all objects and data abstraction is done individually to each object.

Encapsulation: Selective hiding of class members in object. It is supported with keywords private, protected and public called access specifiers.

Polymorphism : One thing in many forms. It is further classified into compile time and runtime polymorphisms.

Inheritance: Getting features of a class into another. In this, the class giving features is called super class and the class getting features is called derived class.

Different benefits of oops achieved by these concepts are better memory management, high data security, easy identification of methods that process data, reusability of code etc.

Q. Explain about oops concepts.

Reading Input with Java.util.Scanner class

The basic IO operations can be done in java using streams. The output can be displayed onto monitor with the method print() and println() of PrintStream class. The PrintStream class is created with a static object within System class, so we can use System.out.print() or System.out.println() to display data onto monitor. The programmers who are much influenced by syntaxes of C, can also use printf() of System.out object very similar to the printf() of C.

The data can be read from keyboard using the System.in object. Instead of using streams, the java.util package has a class Scanner, using which we can read data from keyboard easily. The Scanner class object can wrap the System.in object to read data from keyboard. The Scanner has methods nextInt(), nextFloat(), nextLine() etc methods to read data.
While taking input of a string using nextLine() and if already there is a previous input existing which is not read with nextLine(), then the previous input's ENTER pressed by user is read by nextLine(), which leads to a situation,where user cannot give input of string, if there is a previous input read with other method.

Q. write about reading and displaying data in java

Control Statements

The control structures are used for controlled execution of code. The different control structures in java are : branching using if-else and switch case, and iteration using loops.

The if-else can be used in different ways:

Simple if : the if is a keyword that accepts a condition/Boolean expression for decision making and executes the influenced statement if the condition results true. The else part is optional part. We can use only simple if without else.

syntax :
```
if(condition)
        Statement ;
```

If-else statement : we can also have an else part placed with if keyword. The statement under else part executes if the Boolean expression placed with "if" results false.
Syntax:
```
if(condition)
        Statement;
else
```

Statement;

If we need to place more than one statement under influence of any part, then we should group them using braces.

Nesting if-else statement: placing of if or else with in another if or else is called nested if-else statement. If we need a condition to be checked based on the other if condition then we can place if within if, and it is called nested if statement.

If-else-if ladder: When we need to divide program into more than two parts and execute only one among them, then we use the if-else-if ladder.

```
if(condition)
        Statement;
else if (condition)
        statement;
else
        statement;
```

Iteration :

The loops are used for iteration. Iteration is repeated execution of process. The loops are controlled with a Boolean expression. If the Boolean expression results true, the loops continue its execution and otherwise stops.

Based on placement of the condition, the loops are classified into two types. i) entry controlled loops, ii) exit controlled loops. The loops that have Boolean expression at end of loop body are called entry controlled loops and that have at end of loop body are called exit controlled loops.

While statement:
Syntax :

```
Initialization;
While(boolen expression)
{
Statements;
Counter;
}
```

Do-while : When we use the do-while, the loop body is executed at least once even if the Boolean expression results false.
Syntax :

```
Initialization;
do{
statements;
counter;
}while(Boolean expression) ;
```

for statement:
the for is entry controlled loop and it takes the Boolean expression at beginning of loop body.

Syntax :
Initialization;
for(; Boolean expression;)
{
Statements;
Counter;
}
As in for loop, the program control comes to the location that is before first ; we can logically shift initialization to that location and as every time the program control comes to the location that is after second ; we can logically shift the counter to that location, thus forming logical syntax of for as follows:

```
for(initialization ; condition ; counter)
        Statement;
```

For - each loop:
The for in java takes another form also. It is generally used with arrays etc.
for (type var : array) {
code to be executed
}
Ex:

```
int arr[]={12,23,44,56,78};
for(int i:arr){
System.out.println(i);
}
```

Switch statement:

For decision making, the java supports switch-case statement also. The switch keyword takes a variable and there can be no. of cases grouped with in braces under a switch. The switch checks for EQUALITY of the variable against multiple values. The block under switch executes completely from matching case till end of block. So for controlling, we may place a “break” keyword after statements under each case.
The switch-case in C,C++ work with ASCII code. So a switch can have maximum 256 cases in C and C++. But in java, the switch works with byte, short, int, long data types and from the version of java 7, it works with String objects also. In java’s switch-case we can use WRAPPER classes also.
Syntax :

```
Switch(var)
{
case 1: statements;
break;
case 2: statements;
break;
……..
……..
case n: statements;
break;
Default : statements;
}
```

Q. write about controlling constructs
OR they may ask only branching or only switch case or only iteration for 5 marks
---
Arrays

Array is arranged things. The things are arranged for easy and faster manipulation. The array is collection of similar type of elements that has contiguous memory location. The array is indexed collection of elements. Where the first element is indexed with ZERO and the last element is indexed with its size-1. The "[ ]" are used to declare and manipulate arrays. The [ ] are used to specify "number of elements in array / size of array" while declaring, and the same [ ] are used to specify index number of each element.

The java arrays are dynamic and treated as objects in java. When an array is declared in java, it is allocated with an extra integer variable memory called "length" that stores the size of array and can be used with a DOT operator with array name.

In java array declaration can be done in following ways:
Data_type [] arr_name; ([] near data type)
Data_type []arr_name; ([] near array name)
Data_type arr_name[]; ([] after aray name)

While declaring the array the size is not specified. The size of array is specified while allocating memory with new operator.

Ex: int [] a=new int[5];

In java the arrays can have primitive data type values and also objects as well.
The arrays can be initialized while declaring by specifying the values with in { } directly without specifying array size, as below:
int a[]={10,20,30,40};
int a[]=new int[]{10,20,30,40,50};
int a[]=new int[5]{10,20,30,40,50}; this code raises error.
Ex:
```
int a[]=new int[]{10,20,30,40,50};
for( int i=0;i<a.length;i++)
        System.out.println(a[i]);
```

In java, the "for–each" loop can be used for traversing the across array. The first in for-each is a variable of data type of array and the second is array /collection
name, where each value in the array is copied into the variable and the variable is manipulated while iterating. But as we cannot traverse in reverse in for-each loop, it is fewer used.
Ex:
```
int arr[]={12,13,14,44};
int total=0;
for(int i:arr)
total=total+i;
System.out.println("Total: "+total);
```
We can also pass an array as argument to methods :
```
class Testarray2{
```

```
//creating a method which receives an array as a parameter
static void min(int arr[]){
int min=arr[0];
for(int i=1;i<arr.length;i++)
if(min>arr[i])
min=arr[i];
System.out.println(min);
}
public static void main(String args[]){
int a[]={33,3,4,5};//declaring and initializing an array
min(a);//passing array to method
}}
```

The disadvantage with arrays is that, we can store only fixed size of elements in array. Its size is not increased during runtime of program.

Questions that can be framed:

White about single dimensional arrays of java

How to create and manipulate arrays

Strings-String Class Methods

In programming languages the sequence of characters is called a String. In C and C++, the char data type arrays are used to store and manage strings. In java, the java.lang package has classes String, StringBuffer and StringBuilder to store and manage string data.

The String objects are managed in a special memory location called STRING POOL within the memory. The string pool would not store duplicate values within itself. When the same value is stored in multiple string objects, the data is stored in string pool once and all String references point to the same data in string pool, thus providing better memory management.

When multiple string objects point to same data, if data modified through object then the data of other objects also get modified and it leads to data corruption. So the String objects are designed as IMMUTABLE (NOT MODIFIABLE with in itself). The StringBuffer, StringBuilder classes are mutable (modifiable with in itself) as their data is not managed with in string pool.

In java, when a string literal is represented within " " , it is treated as an instance of String class. So we can initialize the String object either with a new operator or simply assigning string literal to the reference directly.

Ex:

```
String s=new String("HELLO");
String s="HELLO";
```

The string class has number of constructors that take byte[], char[], string constant, string object etc to construct the string object.

The different methods of String class are :

concat() : this method concatenates a given string without modifying data within the object, returns data as a new String object.

Ex: String s= “sri ”;
System.out.println(s);//displays –sri
s.concat(“gnanambica”);
System.out.println(s);//still displays –sri
As the concat() has not modified data within the object s. As these methods return string after doing process, we can store it into another object.
String s1=s.concat(“gnanambica”);
System.out.println(s+” “+s1);//display –sri sri gnanambica

If we need to modify data within itself, then as must store data back into same object as….
s=s.concat(“gnanambica”);

equals() : returns a Boolean value by checking equality of strings

equalsIgnoreCase():returns a Boolean value by checking equality of strings by ignoring its case

toLowercase() : converts string into lower case letters and returns as new string

toUppercase() : converts string into upper case letters and returns as new string

trim() : removes trailing blank spaces

split(char) : splits string into String[] at the char specified.

charAt(int index) : returns the character at given string

startsWith() / endsWith() : for strings searching

compareTo() : for lexicographic comparison of strings
* The String class is overloaded with operator + for string concatenation.

Q. Write about strings in java
Q.How strings are managed with in string pool
