Name : Rajesh Suthar
Roll No :- 50
Sub :- Java programming

PAGE No.
DATE : / /

1] Explain method overloading with suitable example.

→ 1. Method overloading is a feature that allows a class to have more than one method with the same name, if their argument list's are different.

2. It is an important feature in java, there are several cases where we need more than one methods with same name, for example, if we are building an application for calculator, we need different variants of 'add' method based on the user inputs such as add (int, int), add (float, float) etc.

3. It is similar to constructor overloading in Java, that allows a class to have more than one constructor with different argument list's.

```
class addition
{
int add (int a, int b)
 {
  return a+b;
 }
double add (double a, int b)
 {
  return (a+b);
 }
int add (int a, int b, int c)
 {
  return a+b+c;
 }
```

```
public static void main(String args[])
{
addition ob = new addition();
System.out.println(ob.add(20, 30));
System.out.println(ob.add(20.5, 30));
System.out.println(ob.add(20, 30, 40));
}
}
```

2] write a short note on static members

-> 1] Static members are data members (variables) or methods that belong to a static or a non static class itself, rather than to objects of the class.

2] Static members always remain the same, regardless of where & how they are used.

3] Static members are associated with the class, it is not necessary to create an instance of that class to invoke them.

4] A static member can be declared using access control modifiers.

5] A static members has access to all static members of it's containing class, including private members.

6] static member classes & interfaces can be defined only within top-level classes & other static member classes & interfaces.

```
class Student
{
int rollno;
String name;
static String college ="CUG";
Student(int r,String n){
rollno = r;
name = n;
}
void display()
{
System.out.println(rollno+" "+name+" "+college);
}
public static void main(String args[])
{
Student s1=new Student(111,"Karan");
Student s2=new Student(222,"Aryan");
s1.display();
s2.display();
}
}
```

3] write a short note on Finalize method.

→ 1] Finalize method in Java is an object class method that is used to perform cleanup activity before destroying any object.

2] It is called by garbage collector before destroying the object from memory.

3] Finalize() method is called by default for every object before it's deletion.

4] The time at which the garbage collector calls finalizers is dependent on the JVM's implementation & the system's conditions, which are out of our control..

5] Finalize() method is a protected & non-static method of java.lang.Object class, this method will be available in all objects you create in java.

6] It is the method that is used to perform some final operations or clean up operations on an object before it is removed from the memory..

```
public class main
{
int instanceVariable1 = 10;
final int instanceVariable2 = 20;
void showValues()
 {
 System.out.println("instanceVariable1:" +
                                    instanceVariable1);
System.out.println("instanceVariable2:" +
                                    instanceVariable1);
instanceVariable1 = 12;
instanceVariable2 = 30;
System.out.println("instanceVariable1:" +
                                    instanceVariable1);
System.out.println("instanceVariable2:" +
                                    instanceVariable1);
}
public static void main(String args[])
 {
```

```
Main main=new Main();
main.showValues();
}
}
```

4] write a short note on Final class, Final variables & methods.

-> 1] Final :- variables :-

1] The final keyword in java is used to restrict the user. The java final keyword can be used in many context, Final can be
• variable • method • class.

2] The final keyword can be applied with the variables, a final variable that have no value it is called blank final variable or uninitialized final variable.

3] It can be initialized in the constructor only

4] The blank final variable can be static also which will be initialized in the static block only.

If you make any variable as final, you cannot change the value of final variable (It will be constant)

```
class Bike9
{
Final int speedlimit=90;
void run()
{
speedlimit=400;
```

```
}
public static void main(String args[])
{
Bike9 obj = new Bike9();
obj.run();
  }
}
```

2] Final method:

If you make any method as final, you cannot override it.

```
class Bike
{
final void run()
{
System.out.println("running");
}
}
class Honda extends Bike
{
void run()
{
System.out.println("running safely with
                    100Kmph");
}
public static void main(String args[])
{
Honda honda = new Honda();
honda.run();
}
}
```

2] Final method: -

If you make any method as final, you cannot override it.

3] Final class: -

If you make any class as final, you cannot extend it.

```
final class Bike
{ }
class Honda1 extends Bike
{
void run()
{
system.out.println("running safely with 100kmph");
}
public static void main(String args[])
{
Honda 1 honda=new Honda1();
honda.run();
}
}
```

5] write a short note on Garbage collection in Java.

→ 1] Garbage collection is a process which is performed for memory management.

2] It is used for reclaiming the runtime unused objects

3] Java programs compile to bytecode that

can be run on a Java virtual Machine, or JVM for short.

4] when Java programs run on the JVM, objects are created on the heape, which is a portion of memory dedicated to the program.

5] Eventually, some objects will no longer be needed.

6] The garbage collector finds these unused objects & delets them to free up memory.

7] Java garbage collection is an automatic process, the programmer does not need to explicity mark objects to be deleted.

8] The garbage collection implementation lives in the JVM (Java virtual Machine).

```
class stud
 {
 stud()
 {
 System.out.println("Hello student");
 }
 }
 public class nc2
 {
 public static void greet()
 {
 stud obj = new stud();
 }
 public static void main(string args[])
 {
  int no_of_students = 5;
```

```
for(int i=0; i<no_of_students; i++)
  {
    Greet();
  }
 }
}
```

[6] What are various types of constructors in JAVA.

-> 1] Java constructors or constructors in JAVA is a terminology is a been used to construct something in our programs.
2] A constructor in JAVA is a special method that is used to initialize objects.

1] Default constructor:-
  1] If we do note create any constructor, the JAVA compiler automatically create a no-arg constructor during execution of the program.
  2] This constructor is called default constructor.

```
class Main
{
int a;
boolean b;
public static void main(String args[])
 {
 Main obj=new Main();
 System.out.println("Default value:");
 System.out.println("a="+obj.a);
```

```
System.out.println("b= "+obj.b);
    }
}
```

6] Parameterized constructor:

A Java constructor can also accept one or more parameters. Such constructors are known as parameterized constructors (constructor with parameters).

```
class Main
 {
 String Languages;
 Main(String lang)
  {
  languages =lang;
  System.out.println(languages + "programming
                                  language");
  }
  public static void main(String args[])
  {
  Main obj1 = new Main("Java");
  Main obj2 = new Main("Python");
  Main obj3 = new Main("C");
  }
  }
```

7] what is class & what is difference between JAVA class & C++ class

->

Parameter	C++	Java
Input mechanism	I/O statements use in & cout, e.g., in>> x; cout<<y;	I/O statements use (system.in); output is easy, e.g System.out.println(x);
Compiler & Interpreter	C++ only support compiler	Java supports both compiler & intepreter
Concept	write once compile anywhere	write once run anywhere everywhere
Memory management	Accessible to programmer	System controlled
Best features	C++ supports object-oriented features procedural programming features	Java support automatic garbage collection. It does not support destructors.
programmer responsibility	System Responsibility	Runtime error deetection
structure	It supports structures	It doesn not any support for structures.

2. what is object. write a program to create object.

→ object:-

1] A Java object is a member (also called an instance) of a Java class.

2] Each object has an identity, a behavior & a state.

3] The state of an object is stored in fields (variables), while methods (functions) display the object's behavior.

4] objects are created at runtime from templates, which are also known as classes.

```
public class Main
{
int x=5;
public static void main(String args []) {
Main obj =new Main();
System.out.println(myobj.x);
}
}
```