Assignment - 6.

Q.1. What a is method overloading in java and explain with an example?

→ Method overloading in java means having two or more methods in a class with same name and different parameters. It can be with a different parameters or different data types of parameters.

Example -

```
void fun(double n) { ... }
void fun (int a, int b, double c) { ... }
int ~~void~~ fun (int a) { ... }
~~void~~ double fun (int a, float b) { ... }
```

In above example, fun() is overloaded using a different number of parameters and different data type of parameters.

Q.2. What are the rules for method overloading resolution in Java? How does Java determine which overloaded method to call?

→ Rules :-

1) The number of parameters must be different.
2) The data type of parameters must be different
3) The sequence of parameters must be different.

If the above rules are not met, then the compiler will give an error.

When an overloaded method is invoked, java uses the type, number, and sequence of arguments to determine which version of the overloaded method to use

Q 3. What does the static keyword mean in Java? Explain the difference between static and non-static methods.

→ Static is modifier in java. The static keyword in java is used to declare class variable and method. Static fields and methods are called using class name.

Static Method	Non-static Method.
1) A static method belongs to a class, but not to an instance of that class.	1) A non-static method belongs to an instance of a class.
2) A static method can be called without creating an instance of the class.	2) A non-static method can only be called after creating an instance of the class.
3) Static methods are allocated in the method area of the JVM.	3) Non-static methods are allocated in the heap memory
4) The main() method is a static method.	4) ~~Ay~~ Any method that is not declared as static is a non-static method.

Q. 4. Can static methods be overloaded and overridden in Java? ~~Explain the difference between static and non-static methods.~~ How are static variables shared across multiple instances of a class?

→ Static methods in java can be overloaded but not overridden.

Static variables are shared across multiple instances of a class because they are stored in the class itself, rather than in each individual instances. This means, that when you change the value of a static variable, the change is reflected for all instances of the class.

Rainbow

Q. 5. What is the role of the static keyword in the context of memory management.

→ In java, the static keyword is used for memory management. It's used to share a variable or method across other classes. Static variables are shared among all instances, reducing the amount of memory required. It helps optimize memory usage, manage data sharing between function calls and instances, and encapsulate functionality in a modular and organized manner.

Q. 6. What are the significance of the final keyword in Java?

→ The final keyword is a non-access modifier used for classes, attributes, and methods, which makes them non-changeable ₹. We cannot even inherit or override.

Eg. Pie (π) value.

Significance :-

① Preventing modification

② Enhance Secuority.

③ Documenting code (certain variables are to be constant).

When any method is final that means method cannot be overridden by a subclass.

When any class is final that means class cannot be extended.

Q.7. Can a final method be overridden in a subclass? How does the final keyword affect variables, methods and classes in Java?

→ No, a final method cannot be overridden in a subclass.

Variable:- When a variable is declared as final, its value cannot be changed once it has been initialized. It behaves as a constant.

Methods:- When a method is declared as final, it cannot be overridden by any subclass.

Classes:- When a class is declared as final, it cannot be extended by any subclass.

Q.8. What does this keyword represent in Java? How is this keyword used in constructors and methods?

→ The this keyword in Java refers to the current object. The most common use of this keyword is to eliminate the confusion between class attributes and parameters with the same name. It can be used within constructors and methods to access or modify the fields of the current object.

Usages:-

① Constructors:- To call another constructor in the same class, use this(). keyword. This is called as explicit constructor invocation.

② Methods:- To differentiate between class attributes and parameters with the same name, use the this keyword followed by the field name.

Rainbow

Q. 9. What are narrowing and widening conversions in Java?

→ Widening :- Widening conversion is an automatic or implicit conversion that converts a smaller data type to a larger data type. No data loss in widening conversions.
E.x. converting a byte into int.

Narrowing :- Narrowing conversion is a manual or explicit conversion that converts a larger data type to a smaller data type. A Data loss occure in Narrowing conversions.
Ex. Converting into into a byte.

Q. 10. Provide examples of narrowing and widening conversions between primitive data types.

→ ① int x = 40;
byte b = (byte) x;
SOP (b);

② int x = 40.
double d = x;
SOP (d);

③ short s = 100;
int a = s;
SOP (a);

④ Float s = 15.05f.
double d = s;
SOP (d);

⑤ long l = 1000;
int x = ~~to~~ int(l);
SOP (x);

Q 11 How does Java handle potential loss of precision during narrowing conversions?

→ Java handle potential loss of precision during narrowing conversions by explicity cast, the value to the smaller data type.

Ex., if you have long variable and you want to assign it to an int variable, you must explicitly cast the long variable to an int. This is because the int data type has a smaller range value than long data type, and there is possibility of lossing precision in the conversion.

```
long l = 1000;
int x = (int) l;
```

Q.12. Explain the concept of automatic widening conversion in Java.

→ Automatic widening conversion in Java is a feature that allows the compiler to automatically convert a value of a smaller data type to a value of a large data type. This is done with without memory loss.

Rules to be followed :-

① The target data type must be larger than the source type.

② The two data types must be compatible.

Q13. What are the implications of narrowing and widening conversions on type compatibility and data loss?

→ Widening conversions preserve the source value but can change its representation. Widening conversions change a value to a data type that can have any possible value of the original data, while narrowing conversions change a value to a data type that might not be able to hold some of the possible value. For example, converting from an integral type to Decimal, or from Char to String, is a widening conversion. Widening conversions has less prones compare to narrowing as widening has safer transation between data types.