Date: / /

Assignment No:-6

Q) What is method overloading in Java ? Explain with an example?

→ Method overloading have ability to have multiple methods in same class with the same name but different parameters. These method can have different number or type of parameter.

Eg:-

```
public class Calculator{
    public int add(int a, int b){
        return a+b;
    }
    public int add(int a, int b, int c)
    {
        return a+b+c;
    }
    public String add(String a, String b){
        return a+b;
    }
    public static void main(String args[]){
        Calculator calc = new Calculator();
        System.out.println("Sum of two numbers are :" + calc.add(5,10));
        System.out.println("Sum of three numbers are :" + calc.add(5,10,15));
        System.out.println("Concatenation of 'Hello' and 'World': " + calc.add("Hello","World"));
    }
```

(Q2) What are the rules for method overloading resolution in Java? How does Java determine which overloaded method to call?

→ Rules for method overloading:

① Number of parameters:- It checks number of parameters in the method call. It selects the method with the same number of parameters as in call

② Parameter Types:- If there are multiple methods with the same number of parameters, then compares the parameter types. It select method with parameter that match the types in method call. If an exact match is found, that method is chosen.

~~③ Parameter Type:-~~

③ Java determines which overloaded method to call based on the number of parameter, parameter types, etc.

(Q3) What does the static keyword mean in Java? Explain the difference between static and non-static methods

→ Static keyword is used to define a member (method, variable or block) that belongs to class itself.

① Static variables (class variable):- When a variable is declared static inside class it means there is only one copy of variable shared among all

Date: / /

instance of the class.

→ Static method :- It belongs to the class they can be called directly using the class name.

Difference between static & non static methods

① Static methods and variable can be accessed without creating an object of class, while non static methods and variable required an object to be instantiated

② Static methods can only directly call other static methods and variable, whereas non-static methods can access both static and non static methods and variable directly

③ Static methods and are associated with the class and non static methods are associated with instance of the class.

④ Can static methods be overloaded and overridden in Java & How are static variable shared across multiple instance of class?

→ Yes static methods be overloaded but not overridden in Java

→ Static variables also known as class variable are shared among across all instance of a class. This means that there is only one copy of a static variable, and all instance of the class shared and modified the same variable. If one instance changed the value of a static variable, one then the new value visible to other instance and remain same across class

Date: / /

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

→ ① Static variable :→ Memory allocation
Static variables are allocated memory in a special area known as Method Area or class Area. This memory is shared among all instance of the class

Shared memory :- Since static variable are shared among all instance of class they help conserve memory by not duplicating same variable for each object instance

② Static Method :- Static methods are also stored in method area of memory along the static variable. They do not require object instance to be called and are associated with the class.

③ Role :- By using static variable and method you can avoid unnecessary memory assumption that would occur if each instance of class had it own copy of variable or method.

Static variable help in sharing common data among multiple instance, reducing memory usage compared to instance variable that are unique to each object.

Date: / /

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

→ ① Final variable (constant) :- When final is applied to variable it states that variable value cannot be changed once it is assigned to value.
Final variable must be assigned initialized either at the time of declaration or in a constructor.

② Final method :- When applied final to method so it prevents method from being overridden by subclass.

③ Final class :- When applied final to class it prevents class from being subclassed.

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

→ ① Final variable → If final is used with variable means their value will be unchanged
② If final variable is assigned a value when it is declared, that value can be remains constant throughout the code
③ Final method: when applied to to method so it prevents method from being overridden by subclasses
④ Final class :- When applied final to class it prevents class from being subclasses.

Date: / /

Q.8 What does the this keyword represent in Java? How is this keyword used in constructor & methods?

→ ① This keyword is a reference to the current object within method and constructor. It is used to refer to instance variable and methods of current object.

② In constructor :→ When used in a constructor this refer to current object initialized by constructor. It is used to differentiate between parameters passed to constructor and instance variable of class if they have same name.

③ In method :- When instance method this refer to the current object on which method is invoked. It is used to access instance variable, call other methods of object.

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

→ ① Narrowing and widening conversions refer to the process of converting one data type to another, either with a small range of value to a large range of values. These are handled by compiler.

Date: / /

① Widening Conversion (Implicit conversion):⇒
→ It occurs when a data type of small range is converted to data type with large range
→ It is also also known as implicit conversion because it does not requires any explicit casting.

② Narrowing :- (Explicit Conversion) :-
→ It occurs when a data type with large range is converted to small data type with small range
- It is also known as explicit conversion because it requires to explicitly cast the values to desired data type.

Q.10 Provide example of narrowing and widening conversion between primitive data types
→ Eg:→ ① Widening Conversion:-
↳ converting from a smaller data type to larger data type
int num1 = 10;
long num2 = num1;

② Widening Conversion (Implicit)
- Converting smaller to larger data type automatically by compiler
→ byte num1 = (10)
int num2 = byte

Narrowing Conversion

Date: / /

→ Converting from larger data type into smaller data type, which may result in data loss or truncation

```
double num1 = 10.5;
int num2 = (int) num1;
```

→ Narrowing conversion (Explicit)
* Converting larger data type into smaller data type using explicit casting.

Eg – long num1 = 1000;
int num2 = (int) num1

(Q11) How does Java handle potential loss of precision during narrowing conversion

→ ① Java handle potential loss of precision during narrowing conversion by using explicit conversion casting. When we perform a narrowing conversion from larger data type into smaller data type such as from long to int. you may lose information or encounter truncation because larger data type may not be able to represent all possible values of smaller data type.
② To handle this require explicit casting

Eg long num1 = 1000;
int num2 = (int) num1;

(int) – used to explicitly cast long to an int. This can result in a loss of precision if values is too large. So by explicit casting, the potential loss of

Date: / /

precision and make a ~~one time~~ conscious decision to perform conversion

(Q.11) Explain the concept of automatic widening conversion in Java?

→ Automatic widening conversion in Java refers to implicit conversion of smaller data type to larger data type without requiring explicit casting. This conversion is performed by the Java compiler when there is no risk of losing information or precision.

- The Java support automatic widening conversion for primitive data types in a predefined hierarchy, when each data type occupies more memory and has a wider range of values than it smaller counterparts.

Eg int num1 = 10;
long num2 = num1;

→ int num is an int variable has 10 value. when it assigned to long num2 which is long variable so Java perform ~~explicit~~ widening conversion.

(Q.12) What are the implication of narrowing & widening conversion on type compatibility and ~~ate~~ data will ~~be~~

→ (1) Type compatibility :→ Widening conversions are safe and compatible because they involve converting from a smaller data type to larger data type. The larger data

Date: / /

type can represent all possible values of the smaller type.

Data loss:- There is no data loss in widening conversions because the larger data type can hold all values of the smaller type without truncation or loss of precision.

(2) Narrowing Conversion:→ Type Compatibility:- Narrowing conversion can lead to type compatibility issues. They involve converting from a larger data type to smaller data type which may not be able to represent all values of the larger type.

Data loss:→ Narrowing conversion can result in data loss or truncation. If the value been converted exceeds the range or precision of the smaller data type, it may be truncated leading to loss of information.