CDAC Mumbai PG-DAC MARCH 24

Assignment No- 6

1) What is method overloading in Java & explain with an example.?
- Method overloading means declaring methods with same identifiers but different number of parameters or parameter types.

- E.g.

```
	public class NumberManipulation{
		public static int addNumbers(int num1, int num2){
			return num1 + num2;
		}

		public static int addNumbers(int num1, int num2, int num3){
			return num1 + num2 + num3;
		}

		public static int addNumbers(float num1, float num2){
			return num1 + num2;
		}

	}
```

2) What are the rules for method overloading resolution in Java? How does Java determine which overloaded method to call?
- Rules for method overloading are :
	- Parameters of all the methods can differ in their types
	- Variance in number of parameters in methods

- Java determines which overloaded method to call by checking arguments provided by the user and the corresponding data type and number of parameters declared in class method.

3) What does the static keyword mean in Java? Explain the difference between static and non-static methods.
- Static keyword makes the method or class variable to be called upon, without the need of creating its corresponding object.
- Difference between static and non-static methods are:
	- Static methods can be called upon without the creation of instance
	- Non- static methods needs instance to be created
	- Static methods is stored in method area
	- Non- static methods are stored in heap
	- There can only be one copy of static method for all the instances of that class
	- Multiple copies of same non-static methods are created when instances of

the class is created.

4) Can static methods be overloaded and overridden in Java?How are static variables shared across multiple instances of a class?
- Static methods can be overloaded but not overridden.
- Static variables are created in method area. So, they are not related to instances. Which means whenever a class is loaded, its static variables are also loaded and instantiated to value 0. Hence, its only single copy is created and is shared among all the future instances.

5) What is the role of the static keyword in the context of memory management.
- Static variables and methods are created in method area.So, they are not related to instances. Which means whenever a class is loaded, its static variables are also loaded and instantiated to value 0.

6) What is the significance of the final keyword in Java?
- final keyword makes method unable to be overridden and variables as constant.

7)Can a final method be overridden in a subclass?How does the final keyword affect variables, methods, and classes in Java?
- final method cannot be overridden in a subclass
- final method as stated above makes the method unchangeable
- final variables acts like a constant variable as in we cant assign value to it after its first value
- final classes helps the class to avoid inheritance

8) What does the this keyword represent in Java?How is the this keyword used in constructors and methods?
- this keyword refers to the created instance
- this keyword is used whenever we want to access instance variables to differentiate between user input arguments.

9) What are narrowing and widening conversions in Java?
- Narrowing conversion means type conversion from higher size data type into the lower size data type
- Widening conversion means type conversion from lower size data type into the higher size data type

10) Provide examples of narrowing and widening conversions between primitive data types.
- For conversion from double to int, narrowing is used
- For this, we will have to explicitly mention type conversion
-       double data = 3.1;

```
        int data1 = (int) data;
```

- For conversion from int to double, widening is used
- For this, we dont need to explicitly mention type conversion, it automatically happens

```
        int data = 5;
        double data1 = data;
```

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

12) Explain the concept of automatic widening conversion in Java.
- Automatic widening conversion works because the target data type is more in size
- Hence, java doesnt need explicit type and it automatically does it for use.

13) What are the implications of narrowing and widening conversions on type compatibility and data loss?
- Widening conversions are usually accurate since the memory of the target data type is more than the data type which needs to be converted.
- Narrowing conversion can lead to data loss and incorrect values if the value of the data is very high or very low.
- Also, we cant cast boolean to a number and vice versa.