1. Consider following class:

```
class ConnectionConfig
{
    String driver;
    static ConnectionConfig connectionConfig;
    private ConnectionConfig(){}
    static ConnectionConfig getConnectionConfig(){
        if(connectionConfig==null)
            connectionConfig=new ConnectionConfig();
        return connectionConfig;
    }
    void setDriver(String driver)
    {
        this.driver=driver;
    }
    String getDriver(){
        return driver;
    }
}
```

What will be the output of below code segment:

```
public class Tester {
    public static void main(String[] args) {
        ConnectionConfig config1=ConnectionConfig.getConnectionConfig();
        config1.setDriver("Oracle");
        ConnectionConfig config2=ConnectionConfig.getConnectionConfig();
        config2.setDriver("MySQL");
        System.out.println(config1.getDriver());
        System.out.println(config2.getDriver());
    }
}
```

2. Consider the following code:

```
interface Processor {
    String process(String consumer);
    String process(Supplier<String> supplier);
}
class Test{
    public static void main(String[] args) {
        Processor processor=(s)-> "hello "+s;
        System.out.println(processor.process("how r u?"));
    }
}
```

What will be the output of above code? Give explanation to support your answer.

3. When to use comparator and when comparable?
4. Find below classes:

```
class Address{
    private int code;
    private String city;
}
```

```
class Employee{
    private int empCode;
    private String empName;
    private Address address;

}
```

Give definition of following method assuming, all the required getters and setters are available:

void printEmployeeInAscendingOrderOfCity(Employee[] emps)

{

```
        //write logic here
}
```

5. Given following code:

```
        interface TakeRest
        {
            String printSomething(String x);

        }
```

```
public class TestMethodReference {

    public void greet(String userName){
        System.out.println("Have a good day!"+userName);
    }
    public static void main(String[] ar)
    {
    TestMethodReference testMethodReference=new TestMethodReference();

         TakeRest doRelax=testMethodReference::greet;

         doRelax.printSomething("Alex");
}
```

Will the above code print **Have a good day! Alex**? Explain why?

If not, what is the issue?

6. Consider following class:

```
    public class Trainee{
        private int id;
        private String name;
       }
```

Assuming all getters,setters ,constructor are available.
Consider following method in Main class:

```
    static int countNull(Trainee[] trainees, Predicate<Trainee> predicate){
    int nullCount=0;
        for(Trainee trainee:trainees)
        {
            if(predicate.test(trainee))
                nullCount++;
        }
        return nullCount;
    }
```

Write statements to invoke above method in main()

7. What is the cause of ClassCastException? Give code segment.
8. What changes you'll make in Sample class given below to make it immutable?

```
        class Sample
        {
            private int sampleId;
            Sample(int sampleId)
            {
                this.sampleId=sampleId;
            }
            public int getSampleId(){
                return sampleId;
            }
            public void setSampleId(int sampleId)
            {
                this.sampleId=sampleId;
            }
        }
```

9. What are the benefits of using design patterns in application?
10. What is the difference between lambda expression and anonymous class?