1)NullPointerException

2) An exception is an unwanted or unexpected event, which occurs during the execution of a program i.e at run time, that disrupts the normal flow of the program's instructions.

We can handle exceptions using try catch blocks.

We can handle exception using finally block.

We can handle exception using custom exceptions.

3)Creating our own exception by extending Exception class is called custom exception.

We use it to customize the exception according to our desire.

4)Wrapping the data and code acting on the data together.

Rules:

1.Class must be public

2.class must not be final

3.Class should have public default constructor

4.Variables should be private

5.Variables should have getters and setters

5)Ability of an object to take on many forms.

Types:

1.compile time

2.run time

6) Using same method name with different type of arguments and no of arguments.

```
package com.thought focus.Morning.Overloading;

public class Friend {
```

```
}

package com.thought focus.Morning.Overloading;

public class Unknown {

}

package com.thought focus.Morning.Overloading;

public class Human {

    public void speak(Friend friend) {
        System.out.println("You speak casually");

    }

    public void speak(Unknown unknown) {
        System.out.println("You speak formally");

    }
}

package com.thought focus.Morning.Overloading;

public class Overloading {
```

```
    public static void main(String args[]) {
        Friend friend = new Friend();
        Unknown unknown = new Unknown();

        Human adarsh = new Human();

        adarsh.speak(friend);
        adarsh.speak(unknown);

    }

}
```

7)changing the implementation of the parent class method in the child class. Method signature must be same.

```
package com.thought focus.Morning.Overriding;

public class Father {
    public void drive() {
        System.out.println("Father teaches son
to drive at 50km/hr");
    }

}
```

```
package com.thought focus.Morning.Overriding;

public class Son extends Father {
    @Override
    public void drive() {
        System.out.println("Son drives at
70km/hr");
    }

}
```

```
package com.thought focus.Morning.Overriding;

public class Overriding {
    public static void main(String[] args) {
        Son son = new Son();

        son.drive();
    }

}
```

8)Output:args

9)Duplicate method error

10)a class cannot implement an another class

11)Hiding the implementation fro the user and allowing him to use functionalities.

2 ways:

1.abstract class

2.Interface

Through interface we can achieve 100 percent abstraction.

12) we can initialize the value in
1.litteral way
2.By using Object reference
3.By using constructors
4.By using methods

Example:

```
package com.thoughtfocus.assesmenttwo.initialization;


public class Initialization {
int age;
String name;
```

```
long phoneNumber;
String dateOfBirth="05 may,1960";//Litteral Way


//By Using Constructors
public Initialization(String name, long phoneNumber) {
this.name = name;
this.phoneNumber = phoneNumber;
}


//Initializing local variable in Method
public void initializeValue(String address) {
address="Sirsi";
System.out.println(address);
}


public static void main(String[] args) {
Initialization initialize=new Initialization("Adarsh",9483807068l);
initialize.age=23;//Initializing using object reference
System.out.println(initialize.age);
}


}
```