Question2 **java.lang.ClassCastException** is one of the **unchecked exception** in Java. It can occur in our program when we tried to convert an object of **one class type** into an object of **another class type**.

When will be ClassCastException is thrown

- When we try to **cast an object of Parent class to its Child class type,** this exception will be thrown.
- When we try to **cast an object of one class into another class** type that has not extended the other class or they don't have any relationship between them.

Example

```
class Parent {
  String parentName;
  Parent (String n1){
    parentName = n1;
  }
  public void display() {
    System.out.println(parentName);
  }
}
class Childt extends Parentt {
  String childName;
  Child (String n2) {
    super(n2);
    childName = n2;
  }
  public void display() {
    System.out.println(childName);
  }
}
public class Demo {
  public static void main(String args[]) {
    Child ct1 = new Child ("Jai");
    Parent pt1 = new Parent ("Adithya");
    pt1 = ct1;
    pt1.display();
    Parent pt2 = new Parent("Sai");
     Child ct2 = (Child)pt2;
  }
}
```

Output

```
Jai
Exception in thread "main" java.lang.ClassCastException: ParentTest cannot be cast to
ChildTest
  at Test.main(Test.java:30)
```

Question 3: **Dynamic polymorphism** is a process or mechanism in which a call to an overridden method is to resolve at runtime rather than compile-time. It is also known as runtime polymorphism or **dynamic method dispatch.** We can achieve **dynamic polymorphism** by using the **method overriding.**

Method Overriding

It provides a specific implementation to a method that is already present in the parent class. it is used to achieve run-time polymorphism. Remember that, it is not possible to override the **static** method. Hence, we cannot override the main() method also because it is a static method.

Example:

```
class Bird {
      public void fly(){
            System.out. println("Bird is flying");
            }


}

class Parrot extends Bird{
      public void fly(){
            System.out. println("I am  flying");
            }
      public void sing(){
            System.out.println("I am Singing");
            }

}
```

```
public class BirdMain {

        public static void main(String[] args) {
                Bird b1 = new Parrot();
                b1.fly();

                Parrot pr=(Parrot)b1;  //downcasting
                pr.sing();


        }

}
```