java

```
class Example{
int a,b;
public Example()
{
this.a =0;
this.b =0;
System.out.println("Default Constructor : a ="+a+",b="+b);
}
public Example(int a)
{
this.a=a;
this.b=0;
System.out.println("Single parameter Constructor : a ="+a+",b="+b);
}
public Example(int a, int b)
{
this.a=a;
this.b=b;
System.out.println("two parameter Constructor : a ="+a+",b="+b);
}
public void display(){
System.out.println("Display with no parameters: a ="+a+",b="+b);
}
public void display(int a){
System.out.println("Display with one parameters: a ="+a+",b="+b);
}
public void display(int a, int b){
System.out.println("Display with one parameters: a ="+a+",b="+b);
}
public static void staticMethod(){
System.out.println("this is a static method");
}
public static void main(String[] args)
{
Example obj1 = new Example();
Example obj2 = new Example(5);
Example obj3 = new Example(5,10);
obj1.display();
obj1.display(7);
obj1.display(7,14);
Example.staticMethod();
}
}
```

2)#prac-1b
wap to implement the concept of inheritance and method overriding

```
class A
{
void show()
{
System.out.println("base class");
}
}
```

```
class B extends A
{
void show()
{
System.out.println("Derived Class");
}
}
class pr1b
{
public static void main(String[] args)
{
B s= new B();
A s1= new A();
s.show();
s1.show();
}
}
```

2)wap implement the concept of abstract classes and methods

```
abstract class shape
{
public abstract double area();
}
class circle extends shape
{
private double radius;
public circle(double radius)
{
this.radius=radius;
}
//@override
public double area()
{
return Math.PI*radius*radius;
}
}
class pr2a
{
public static void main(String[] args)
{
circle c=new circle(10.0);
System.out.println("circle area is : "+c.area());
}
}
```

3)write a program to define userdefine Exception and raise them as per requirement

```
import java.util.*;
class CustomException extends Exception
{
public CustomException(String message)
{
super(message);
}
}
public class pr3
{
public static void main(String args[])
{
try
{
int age=-20;
// Scanner sc = new Scanner(System.in);
// System.out.println("Enter the age");
if(age<0)
{
throw new CustomException("Age cannot be negative");
}
System.out.println("age"+age);
}
catch(CustomException e)
{
System.out.println(e.getMessage());
}
}
}
```

4)list interface

```
import java.util.*;
class ListDemo {
public static void main(String[] args) {
List fruits = new ArrayList<>();
fruits.add("Apple");
fruits.add("Banana");
fruits.add("Cherry");
fruits.add("Kiwi");
fruits.add("Banana");
fruits.add("Mango");
System.out.println("List of fruits");
for (String fruit : fruits) {
System.out.println(fruit);
}
System.out.println("\n Element at index 2 :" + fruits.get(2));
fruits.remove("Banana");
System.out.println("\n list after removing Banana :" + fruits);
System.out.println("\n list Contained Mango :" +
fruits.contains("Mango"));
System.out.println("\n iterating using listiterator");
ListIterator iterator = fruits.listIterator();
while (iterator.hasNext()) {
System.out.println(iterator.next());
}
}
}
```

4b) set interface

```
import java.util.*;
public class SetDemo1
{
public static void main(String args[])
{
Set fruits = new HashSet<>();
fruits.add("Apple");
fruits.add("mango");
fruits.add("banana");
fruits.add("orange");
fruits.add("Apple");
System.out.println("Set of Fruits :");
for (String fruit : fruits )
{
System.out.println(fruit);
}
System.out.println("\nSet Contains 'Mango' :"+fruits.contains("mango"));
fruits.remove("banana");
System.out.println("\nSet after removing 'banana' :" +fruits);
System.out.println("\nIterating using Iterator :");
Iterator iterator = fruits.iterator();
while (iterator.hasNext())
{
System.out.println(iterator.next());
}
fruits.clear();
System.out.println("\nSet after clearing :" +fruits);
}
}
```

4c)map interface

```
import java.util.*;

public class MapDemo {
public static void main(String args[]) {
Map map = new HashMap<>();
map.put(1, "Apple");
map.put(2, "Banana");
map.put(3, "Orange");
map.put(4, "Mango");
map.put(5, "Grapes");
map.put(1, "cherry");
System.out.println("Map of Fruits :");
for (Map.Entry entry : map.entrySet()) {
System.out.println("Key :" + entry.getKey() + ", Value : " + entry.getValue());
}
System.out.println("\n value for key 2 : " + map.get(2));
map.remove(3);
System.out.println("\n after removing key 3 : ");
for (Map.Entry entry : map.entrySet()) {
System.out.println("Key :" + entry.getKey() + ", Value : " + entry.getValue());
}
System.out.println("\nMap contains key 4 : " + map.containsKey(4));
System.out.println("map contains value 'Banana' : " + map.containsValue("Banana"));
System.out.println("\n Iterating using forEach method :");
map.forEach((key, value) -> System.out.println("Key :" + key + ", Value :" + value));
map.clear();
System.out.println("\nMap after clearing :" + map);
}
}
```

5) student resume

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class StudentResumeForm extends JFrame {
private JLabel nameLabel, addressLabel, phoneLabel, emailLabel, educationLabel;
private JTextField nameField, phoneField, emailField;
private JTextArea addressArea, educationArea;
private JButton submitButton, resetButton;

public StudentResumeForm() {
setTitle("Student Resume Form");
setSize(500, 600);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLayout(new GridLayout(10, 2, 5, 5));

nameLabel = new JLabel("Name:");
addressLabel = new JLabel("Address:");
phoneLabel = new JLabel("Phone Number:");
emailLabel = new JLabel("Email:");
educationLabel = new JLabel("Education:");
nameField = new JTextField();
addressArea = new JTextArea(3, 20);
phoneField = new JTextField();
emailField = new JTextField();
educationArea = new JTextArea(3, 20);
submitButton = new JButton("Submit");
resetButton = new JButton("Reset");

add(nameLabel);
add(nameField);
add(addressLabel);
add(new JScrollPane(addressArea));
add(phoneLabel);
add(phoneField);
add(emailLabel);
add(emailField);
add(educationLabel);
add(new JScrollPane(educationArea));
add(submitButton);
add(resetButton);

submitButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
JOptionPane.showMessageDialog(null, "Resume submitted successfully!");
}
});

resetButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
nameField.setText("");
addressArea.setText("");
phoneField.setText("");
emailField.setText("");
educationArea.setText("");
}
});

setVisible(true);
}

public static void main(String[] args) {
new StudentResumeForm();
}
```

```
}
```

6)write a jdbc program that display data of given table

```
import java.sql.*;

public class DisplayTableData {
public static void main(String[] args) {
String url = "jdbc:mysql://localhost:3306/sycs?useSSL=false&serverTimezone=UTC";
String user = "admin";
String password = "12345";

try (Connection conn = DriverManager.getConnection(url, user, password);
Statement statement = conn.createStatement();
ResultSet resultSet = statement.executeQuery("SELECT * FROM student")) {

int columnCount = resultSet.getMetaData().getColumnCount();

while (resultSet.next()) {
for (int i = 1; i <= columnCount; i++) {
System.out.print(resultSet.getString(i) + "\t");
}
System.out.println();
}

} catch (SQLException e) {
e.printStackTrace();
}
}
}
```