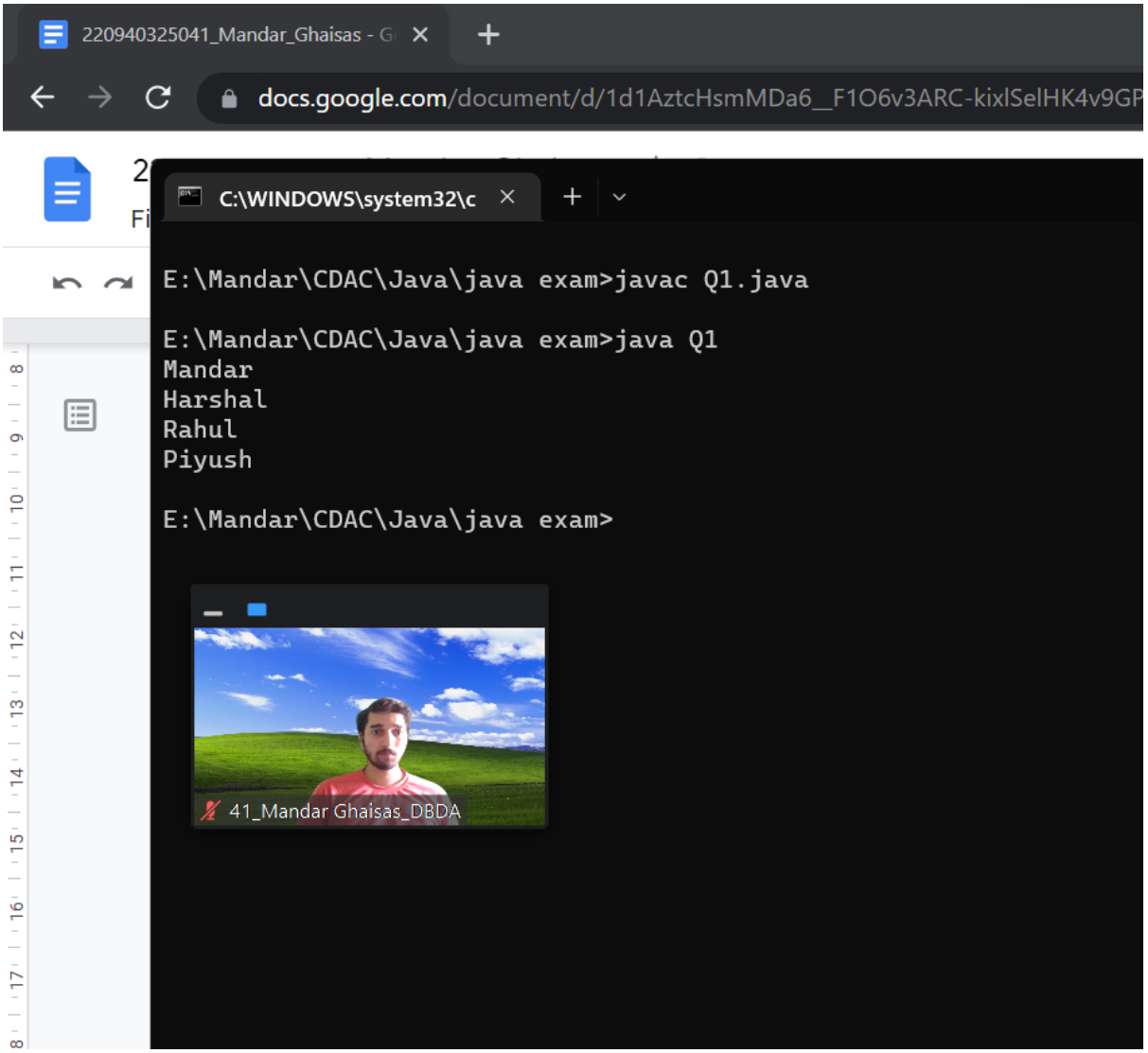
Name - Mandar Ghaisas
Roll No - 220940325041

Q1) Write a Java program to create a new array list, add some elements (string) and print out the collection by using for-each loop.

```
import java.util.*;
public class Q1

{
	public static void main(String args[])
	{
		ArrayList<String> names = new ArrayList<String>();

		names.add("Mandar");
		names.add("Harshal");
		names.add("Rahul");
		names.add("Piyush");
		//System.out.println(names);
		for (String i : names)
		{
			System.out.println(i);
		}


	}


}
```

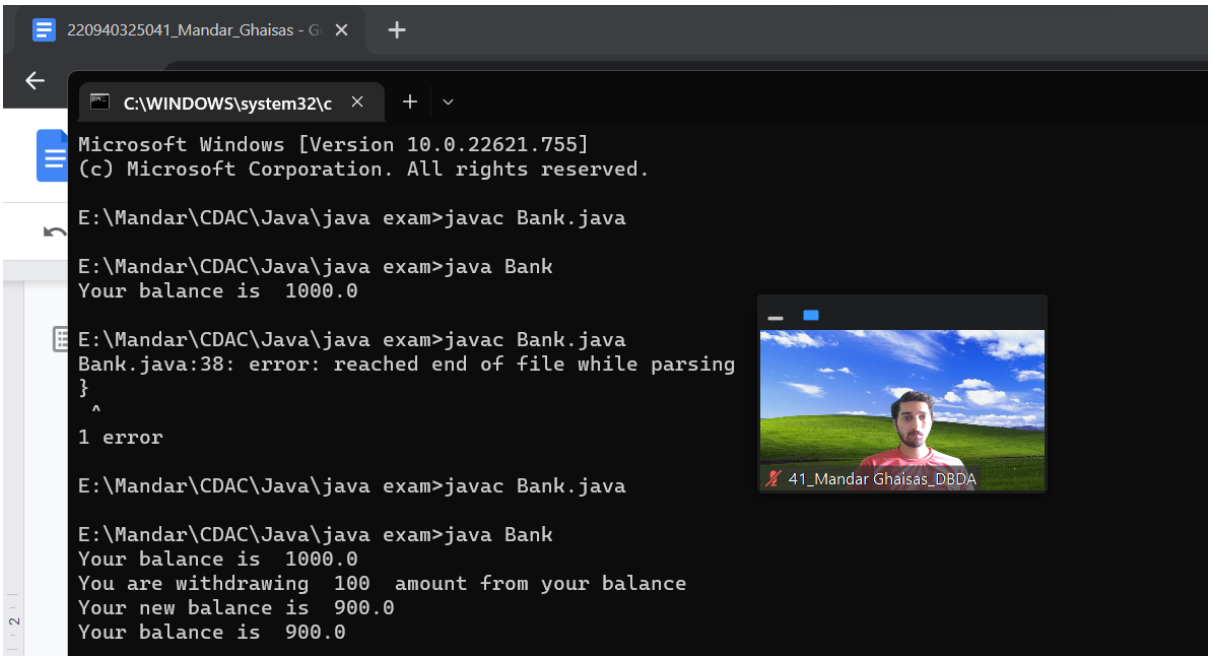

220940325041_Mandar_Ghaisas - G
docs.google.com/document/d/1d1AztcHsmMDa6__F1O6v3ARC-kixlSelHK4v9GP
C:\WINDOWS\system32\c
E:\Mandar\CDAC\Java\java exam>javac Q1.java
E:\Mandar\CDAC\Java\java exam>java Q1
Mandar
Harshal
Rahul
Piyush
E:\Mandar\CDAC\Java\java exam>
41_Mandar Ghaisas_DBDA

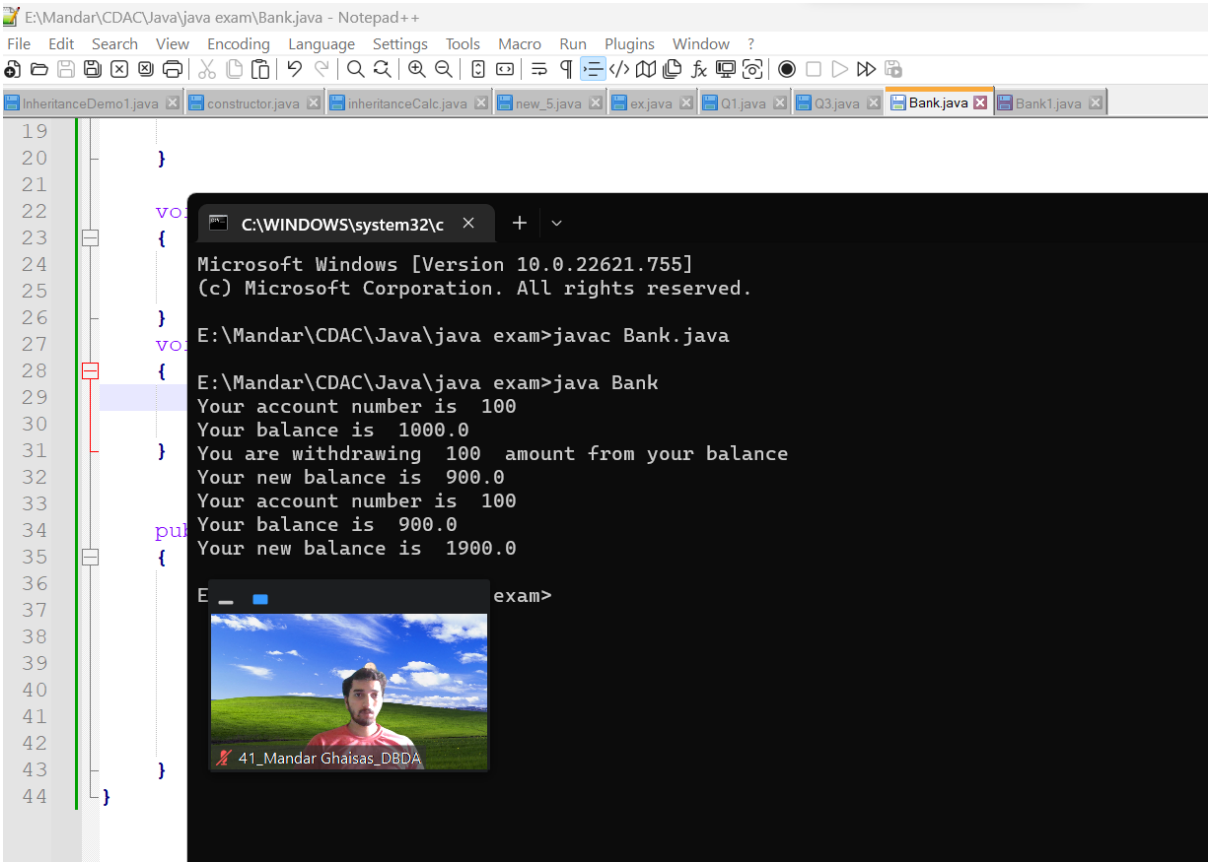
Q2) Develop a class BankAccount having following data members : (10 Marks)

int accno
double balance
Write appropriate constructors to initialize data members
Define the following functions :
withdraw : balance will reduce
deposit : balance will increase
show : display accno and balance
If user tries to withdraw more than the balance, use exception handling code. Demonstrate the
concept of exception handling in main() function.

```
class Bank
{

	int accno = 100;
	double balance = 1000;

	void withdraw(int withdrawal)
	{if (balance> withdrawal);
		try
		{
			System.out.println("You are withdrawing  "+withdrawal+"  amount from
your balance");
			balance = balance - withdrawal;
			System.out.println("Your new balance is  "+balance);
		}
		catch(Exception e)
		{
			System.out.println("You cannot withdraw amount larger than your
balance");
		}
	}

	void deposit(int deposited)
	{
		balance = balance + deposited;
		System.out.println("Your new balance is  "+balance);
	}
	void show()
	{
		System.out.println("Your account number is  "+accno);
		System.out.println("Your balance is  "+balance);
	}


	public static void main(String args[])
```

```
	{
		Bank c = new Bank();
		c.show();
		c.withdraw(100);
		c.show();


	}
}
```

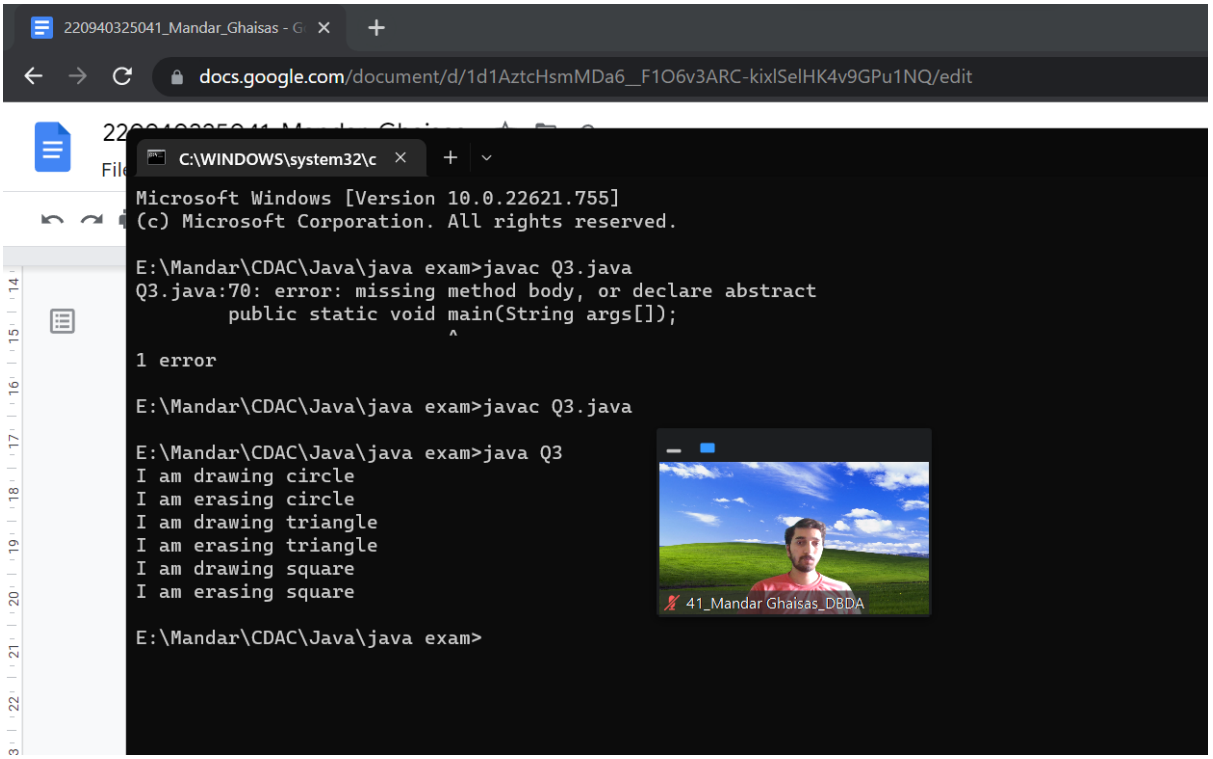



E:\Mandar\CDAC\Java\java exam\Bank.java - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
InheritanceDemo1.java
constructor.java
inheritanceCalc.java
new_5.java
ex.java
Q1.java
Q3.java
Bank.java
Bank1.java
}
C:\WINDOWS\system32\c
Microsoft Windows [Version 10.0.22621.755]
(c) Microsoft Corporation. All rights reserved.
E:\Mandar\CDAC\Java\java exam>javac Bank.java
E:\Mandar\CDAC\Java\java exam>java Bank
Your account number is 100
Your balance is 1000.0
You are withdrawing 100 amount from your balance
Your new balance is 900.0
Your account number is 100
Your balance is 900.0
Your new balance is 1900.0
exam>
41_Mandar Ghaisas_DBDA

Q3) Write a program to create a class named shape. In this class we have three sub classes circle, triangle and square, each class has two member function named draw () and erase (). Create these using Runtime Polymorphism concepts.

```
class Shape
{
	void draw()
	{
		System.out.println("I am drawing a shape");


	}

	void erase()
	{
		System.out.println("I am erasing the shape");

	}

}

class Circle extends Shape
{
	@Override
	void draw()
	{
		System.out.println("I am drawing circle");

	}
	@Override
	void erase()
	{
		System.out.println("I am erasing circle");

	}

}

class Triangle extends Shape
{
	@Override
	void draw()
	{
		System.out.println("I am drawing triangle");


	}
	@Override
```

```
	void erase()
	{
		System.out.println("I am erasing triangle");
	}

}
class Square extends Shape
{

	@Override
	void draw()
	{
		System.out.println("I am drawing square");


	}
	@Override
	void erase()
	{
		System.out.println("I am erasing square");
	}

}
public class Q3
{
	public static void main(String args[])
	{
		Shape c = new Circle();
		c.draw();
		c.erase();

		Shape t = new Triangle();
		t.draw();
		t.erase();

		Shape s = new Square();
		s.draw();
		s.erase();
	}
}
```


220940325041_Mandar_Ghaisas - G
docs.google.com/document/d/1d1AztcHsmMDa6__F1O6v3ARC-kixlSelHK4v9GPu1NQ/edit
C:\WINDOWS\system32\c
Microsoft Windows [Version 10.0.22621.755]
(c) Microsoft Corporation. All rights reserved.
E:\Mandar\CDAC\Java\java exam>javac Q3.java
Q3.java:70: error: missing method body, or declare abstract
public static void main(String args[]);
^
1 error
E:\Mandar\CDAC\Java\java exam>javac Q3.java
E:\Mandar\CDAC\Java\java exam>java Q3
I am drawing circle
I am erasing circle
I am drawing triangle
I am erasing triangle
I am drawing square
I am erasing square
E:\Mandar\CDAC\Java\java exam>
41_Mandar Ghaisas_DBDA

Q4) Constructor chaining

```
class GrandParent
{
	public String grandFathername,grandMothername;

	public GrandParent(String x , String y)
	{
		grandFathername = x;
		grandMothername = y;

		System.out.println("name of grandfather is  "+grandFathername+"  and name
of grandmother is  "+grandMothername);

	}

}
class Parent extends GrandParent
{
	public String FatherName,MotherName;
	public void Parent(String x,String y,String w,String z)
	{
		this(w,z);
		FatherName = x;
		MotherName = y;
		System.out.println("Fathers name is "+FatherName + "Mothers name is
"+MotherName);
	}
	public void Parent(String x,String y);
	{
		super(x,y);
	}
}
public class child extends Parent
{
	child(String x,String y,String w,String z)
	{
		super(x,y,w,z);
	}
	public static void main(String args[])
	{
		child c = new child("Manohar","Madhuri","Manish","Manasi");
	}
}
```

```
(c) Microsoft Corporation. All rights reserved.

E:\Mandar\CDAC\Java\java exam>javac child.java
child.java:15: error: constructor GrandParent in class GrandParent cannot be applied to give
class Parent extends GrandParent
^
  required: String,String
  found:    no arguments
  reason: actual and formal argument lists differ in length
child.java:20: error: call to this must be first statement in constructor
                this(w,z);
                    ^
child.java:20: error: constructor Parent in class Parent cannot be applied to given types;
                this(w,z);
                ^
  required: no arguments
  found:    String,String
  reason: actual and formal argument lists differ in length
child.java:25: error: missing method body, or declare abstract
        public void Parent(String x,String y);
                    ^
child.java:27: error: call to super must be first statement in constructor
                super(x,y);
                     ^
child.java:27: error: cannot find symbol
                super(x,y);
                      ^
  symbol:   variable x
  location: class Parent
```