

Java Module Exam
Roll no :220940325081
Name: Sumit Bansod

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. (10 Marks)

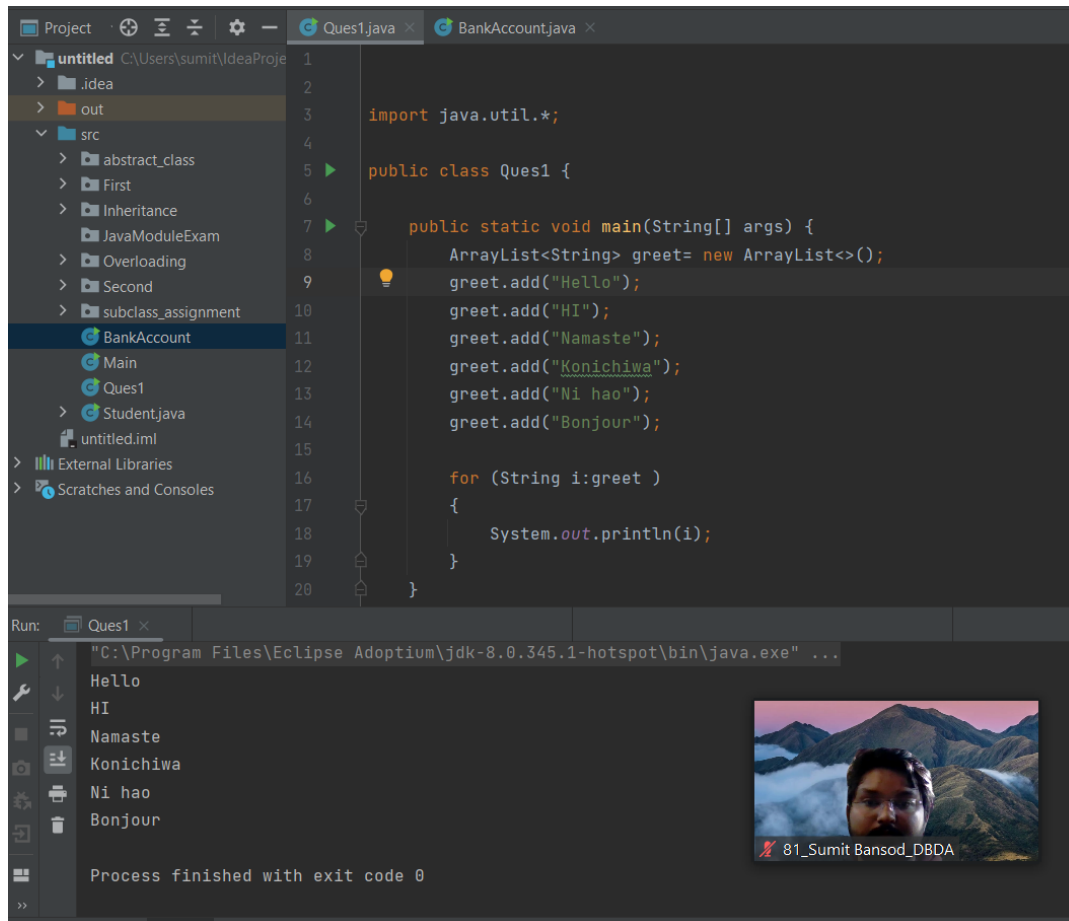
CODE:

```
import java.util.*;

public class Ques1 {

    public static void main(String[] args) {
        ArrayList<String> greet= new ArrayList<>();
        greet.add("Hello");
        greet.add("HI");
        greet.add("Namaste");
        greet.add("Konichiwa");
        greet.add("Ni hao");
        greet.add("Bonjour");

        for (String i:greet )
        {
            System.out.println(i);
        }
    }
}
```



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.

```

import java.sql.SQLOutput;
import java.util.*;

```

```

public class Ques2 {
    public static void main(String[] args) {

```

```

Scanner sc= new Scanner(System.in);
BankAccount i;
System.out.println("Enter the account number : ");
int a=sc.nextInt();

System.out.println("Enter the balance : ");
int b=sc.nextInt();
i=new BankAccount(a,b);
lp :while(true){
    System.out.println("\n1.Withdraw \n2.Deposit \n3.Show Account
Details\n0.EXIT");
    int s= sc.nextInt();
    switch(s){
        case 0:
            break lp;
        case 1:
            System.out.println("Enter the amount to withdraw");
            int g= sc.nextInt();
            i.withdraw(g);
            break;
        case 2:
            System.out.println("Enter the amount to deposit");
            int k=sc.nextInt();
            i.deposit(k);
            break ;
        case 3:
            System.out.println("Loading the balance ");
            i.show();
    }
}
}
}

class BankAccount{
    int accno;
    double balance;
    int amount;

    BankAccount(int a,double b){
        this.accno=a;
        this.balance=b;
    }

    void withdraw(int w){
        this.amount=w;
        try {
            if (balance < amount) {
                throw new ArithmeticException("Enter a valid amount to
withdraw");
            }
        }
    }
}

```

```

    }
    catch(ArithmeticException e){
        System.out.println("Enter A valid amount");
    }
    if (balance>amount)
    {balance=balance-amount;
        System.out.println("The Balance is :"+balance);
    }
}

void deposit(int d){
    this.amount=d;
    balance=balance+amount;
    System.out.println("The balance after deposit is :"+balance);
}

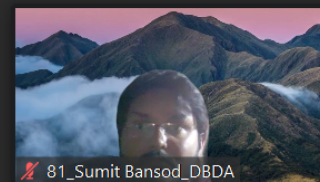
void show(){
    System.out.println("The account number is "+accno+" & the current
balance is "+balance);
}
}
}

```

```

class BankAccount{
    2 usages
    int accno;
    10 usages
    double balance;
    6 usages
    int amount;
    1 usage
    BankAccount(int a,double b){
        this.accno=a;
        this.balance=b;
    }
    1 usage
    void withdraw(int w){
        this.amount=w;
        try {
            if (balance < amount) {
                throw new ArithmeticException("Enter a valid amount to withdraw");
            }
        }
        catch(ArithmeticException e){
            System.out.println("Enter A valid amount");
        }
        if (balance>amount)
        {balance=balance-amount;
            System.out.println("The Balance is :"+balance);
        }
    }
    1 usage
}

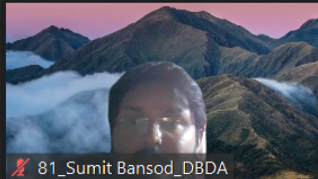
```



```

1 usage
void deposit(int d){
    this.amount=d;
    balance=balance+amount;
    System.out.println("The balance after deposit is :"+balance);
}
1 usage
void show(){
    System.out.println("The account number is "+accno+" & the current balance is "+balance);
}
}
}

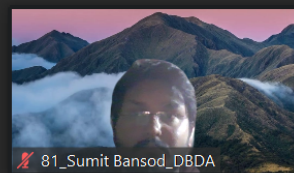
```



```

1 import java.sql.SQLOutput;
2 import java.util.*;
3 public class Ques2 {
4     public static void main(String[] args) {
5         Scanner sc= new Scanner(System.in);
6         BankAccount i;
7         System.out.println("Enter the account number : ");
8         int a=sc.nextInt();
9
10        System.out.println("Enter the balance : ");
11        int b=sc.nextInt();
12        i=new BankAccount(a,b);
13        lp :while(true){
14            System.out.println("\n1.Withdraw \n2.Deposit \n3.Show Account Details\n0.EXIT");
15            int s= sc.nextInt();
16            switch(s){
17                case 0:
18                    break lp;
19                case 1:
20                    System.out.println("Enter the amount to withdraw");
21                    int g= sc.nextInt();
22                    i.withdraw(g);
23                    break;
24                case 2:
25                    System.out.println("Enter the amount to deposit");
26                    int k=sc.nextInt();
27                    i.deposit(k);
28                    break ;
29                case 3:
30                    System.out.println("Loading the balance ");

```



OUTPUT:

C:\Program Files\Eclipse IDE\workspace>cd src\src1\notespot\bin & java.exe -cp ..

Enter the account number :

456879

Enter the balance :

100000

1.Withdraw

2.Deposit

3.Show Account Details

0.EXIT

1

Enter the amount to withdraw

20000

The Balance is :80000.0

1.Withdraw

2.Deposit

3.Show Account Details

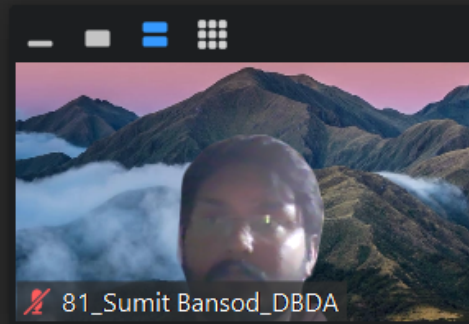
0.EXIT

1

Enter the amount to withdraw

1000000

Enter A valid amount



```
Enter the amount to withdraw
```

```
1000000
```

```
Enter A valid amount
```

```
1.Withdraw
```

```
2.Deposit
```

```
3.Show Account Details
```

```
0.EXIT
```

```
2
```

```
Enter the amount to deposit
```

```
50000
```

```
The balance after deposit is :130000.0
```

```
1.Withdraw
```

```
2.Deposit
```

```
3.Show Account Details
```

```
0.EXIT
```

```
3
```

```
Loading the balance
```

```
The account number is 456879 & the current balance is 130000.0
```

```
1.Withdraw
```

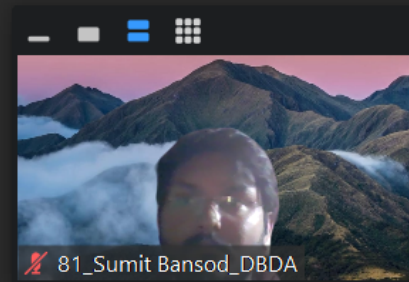
```
2.Deposit
```

```
3.Show Account Details
```

```
0.EXIT
```

```
0
```

```
Process finished with exit code 0
```



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. (10 Marks)

```
public class Ques3 {
```

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

```
        shape myCircle=new Circle();
```

```

        myCircle.draw();
        myCircle.erase();

        shape myTriangle=new Triangle();
        myTriangle.draw();
        myTriangle.erase();

        Square mySquare= new Square();
        mySquare.draw();
        mySquare.erase();

    }
}
abstract class shape{
    abstract void draw();
    abstract void erase();
}
class Circle extends shape{
    @Override
    void draw() {
        System.out.println("The Circle is being drawn");
    }

    @Override
    void erase() {
        System.out.println("The Circle is being erased");
    }
}
class Triangle extends shape{
    @Override
    void draw() {
        System.out.println("The Triangle is being drawn");
    }

    @Override
    void erase() {
        System.out.println("The Triangle is being erased");
    }
}
class Square extends shape{
    @Override
    void draw() {
        System.out.println("The Square is being drawn");
    }

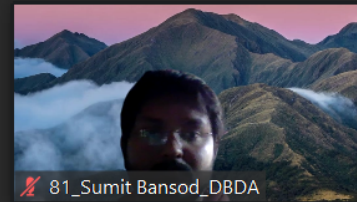
    @Override
    void erase() {
        System.out.println("The Square is being erased");
    }
}

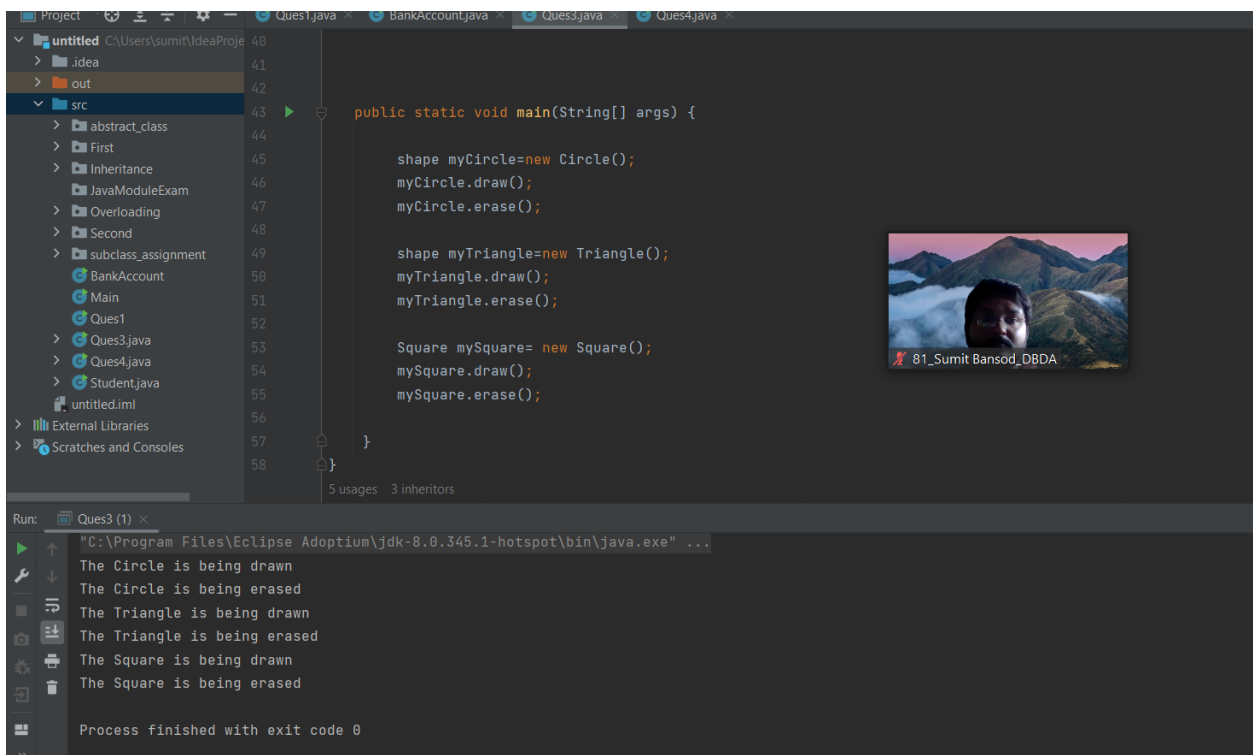
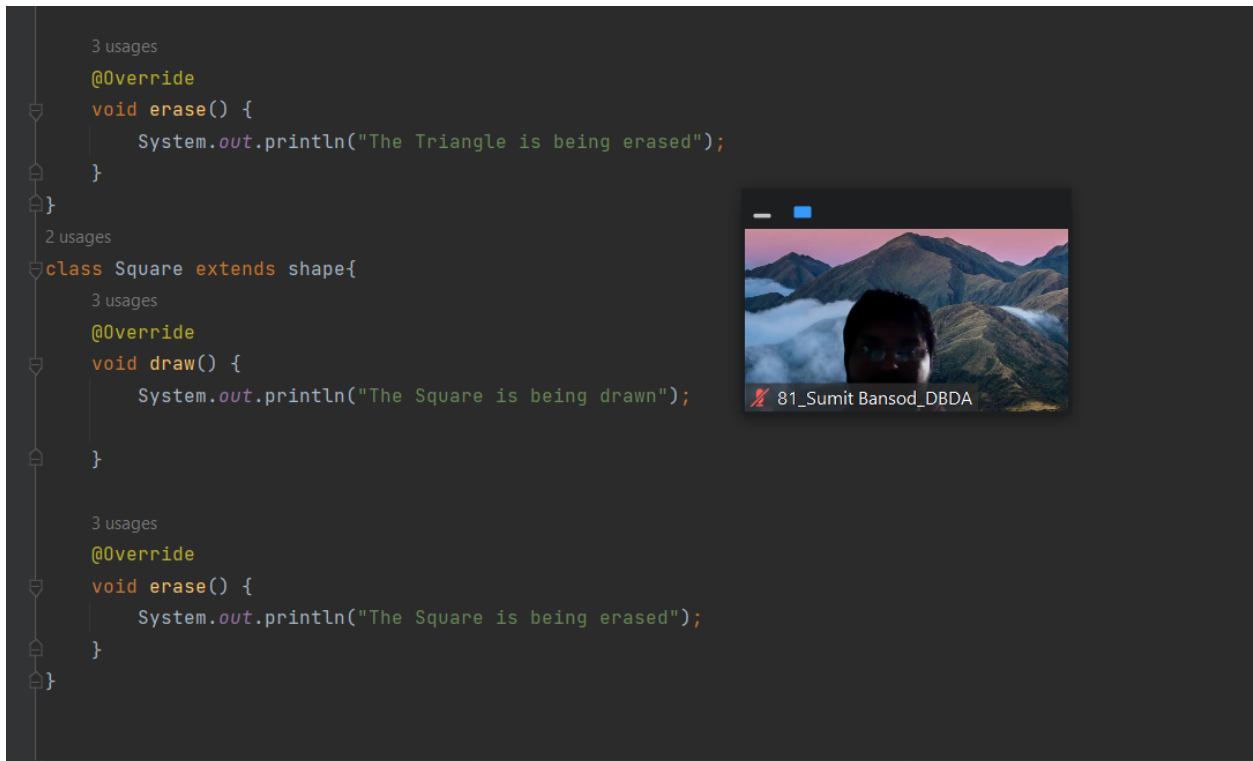
```

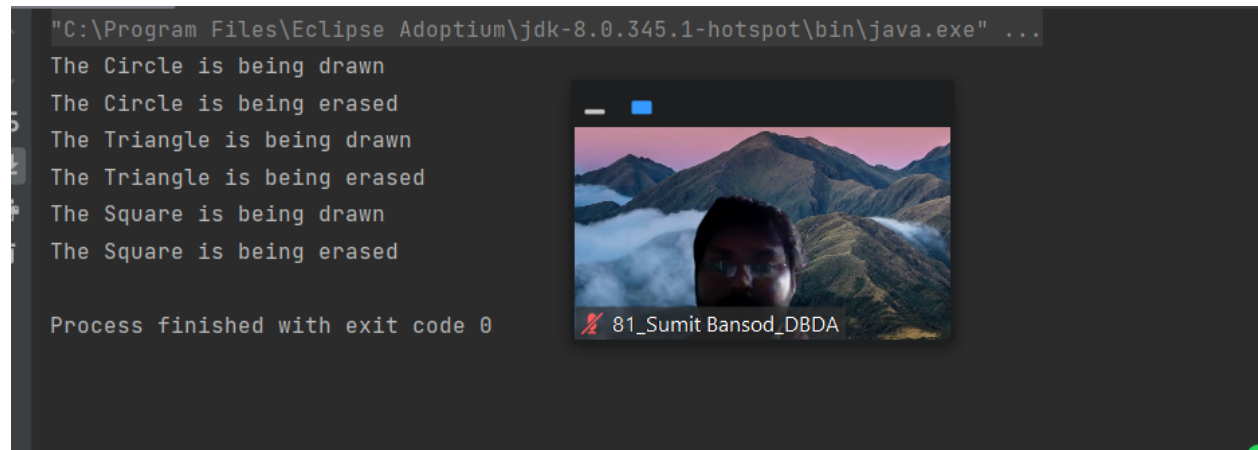


```
}  
}
```

```
3 usages 3 implementations  
abstract class shape{  
    3 usages 3 implementations  
    abstract void draw();  
    3 usages 3 implementations  
    abstract void erase();  
}  
1 usage  
class Circle extends shape{  
    3 usages  
    @Override  
    void draw() {  
        System.out.println("The Circle is being drawn");  
    }  
  
    3 usages  
    @Override  
    void erase() {  
        System.out.println("The Circle is being erased");  
    }  
}  
1 usage  
class Triangle extends shape{  
    3 usages  
    @Override  
    void draw() {  
        System.out.println("The Triangle is being drawn");  
    }  
}
```







Q4.

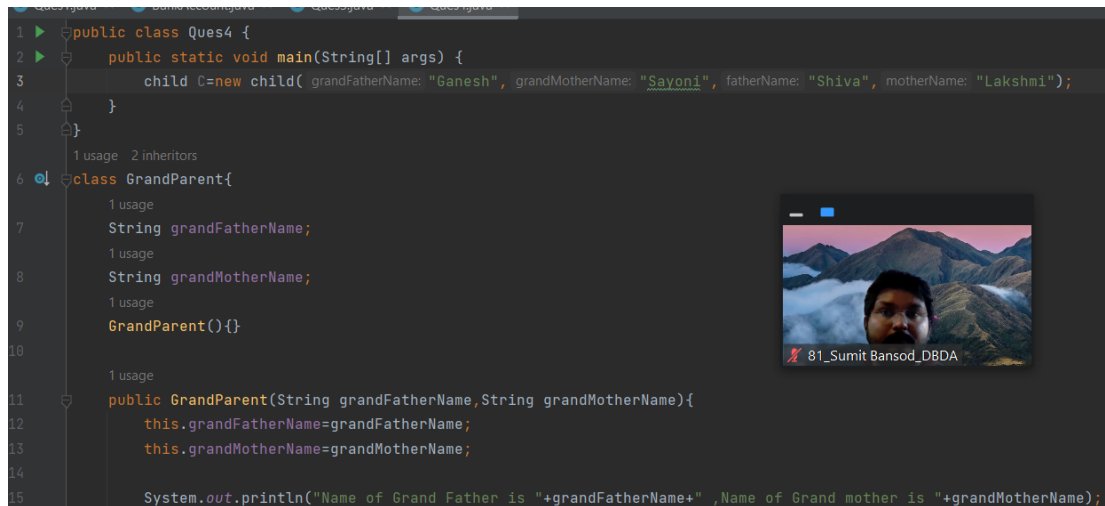
CODE

```
public class Ques4 {  
    public static void main(String[] args) {  
        child C=new child("Ganesh","Sayoni","Shiva","Lakshmi");  
    }  
}  
class GrandParent{  
    String grandFatherName;  
    String grandMotherName;  
    GrandParent(){}  
  
    public GrandParent(String grandFatherName,String grandMotherName){  
        this.grandFatherName=grandFatherName;  
        this.grandMotherName=grandMotherName;  
  
        System.out.println("Name of Grand Father is "+grandFatherName+" ,Name of  
Grand mother is "+grandMotherName);  
    }  
}  
class parent extends GrandParent{  
    String fatherName;  
    String motherName;  
    parent(){}  
    public parent (String fatherName, String motherName, String  
grandFatherName, String grandMotherName) {  
        super(grandFatherName, grandMotherName);  
  
        System.out.println("Name of Father is " + fatherName + " ,Name of mother  
is " + motherName);  
    }  
}  
class child extends parent{
```

```

        public child (String grandFatherName,String grandMotherName,String
fatherName,String motherName){
            super (grandFatherName,grandMotherName,fatherName,motherName);
        }
    }
}

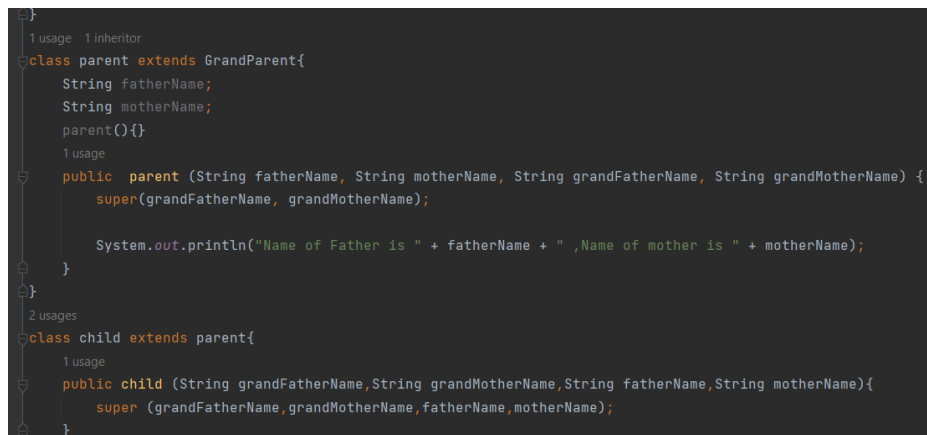
```



```

1  public class Ques4 {
2      public static void main(String[] args) {
3          child C=new child( grandFatherName: "Ganesh", grandMotherName: "Sayoni", fatherName: "Shiva", motherName: "Lakshmi");
4      }
5  }
6  class GrandParent{
7      String grandFatherName;
8      String grandMotherName;
9      GrandParent(){
10     }
11     public GrandParent(String grandFatherName,String grandMotherName){
12         this.grandFatherName=grandFatherName;
13         this.grandMotherName=grandMotherName;
14     }
15     System.out.println("Name of Grand Father is "+grandFatherName+" ,Name of Grand mother is "+grandMotherName);

```



```

1  class parent extends GrandParent{
2      String fatherName;
3      String motherName;
4      parent(){
5      }
6      public parent (String fatherName, String motherName, String grandFatherName, String grandMotherName) {
7          super(grandFatherName, grandMotherName);
8          System.out.println("Name of Father is " + fatherName + " ,Name of mother is " + motherName);
9      }
10 }
11 class child extends parent{
12     public child (String grandFatherName,String grandMotherName,String fatherName,String motherName){
13         super (grandFatherName,grandMotherName,fatherName,motherName);
14     }
15 }

```

untitled C:\Users\sumit\IdeaProje

17 }

18 class parent extends GrandParent{

19 String fatherName;

20 String motherName;

21 parent(){}

22 public parent (String fatherName, String motherName, String grandFatherName, String grandMotherName) {

23 super(grandFatherName, grandMotherName);

24

25 System.out.println("Name of Father is " + fatherName + " ,Name of mother is " + motherName);

26 }

27 }

28 class child extends parent{

29 public child (String grandFatherName,String grandMotherName,String fatherName,String motherName){

30 super (grandFatherName,grandMotherName,fatherName,motherName);

31 }

32 }

Ques4 x

"C:\Program Files\Eclipse Adoptium\jdk-8.0.345.1-hotspot\bin\java.exe"

Name of Grand Father is Ganesh ,Name of Grand mother is Sayoni

Name of Father is Shiva ,Name of mother is Lakshmi

Process finished with exit code 0

81_Sumit Bansod_DBDA