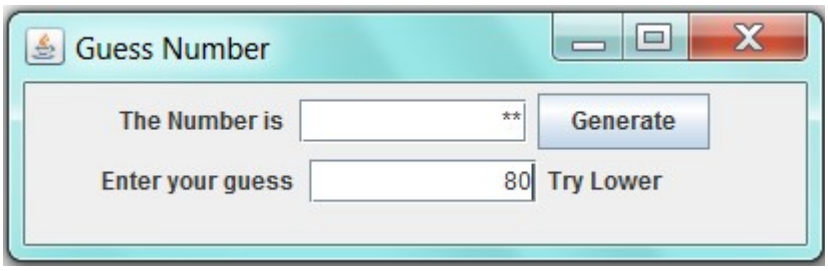
Section 1 (3x18=24 M)

1.

Write a number guessing game in GUI. The program shall generate a random number between 1 to 100. It shall mask out the random number generated and output "You Got it", "Try Higher" or "Try Lower" depending on the user's input.

Hints:

- You can use Math.random() to generate a random number in double in the range of [0.0, 1.0)



2.

Write a Program using Threads for the following case study:

Joint bank account: Write a simple program showing a typical invocation of banking operations (withdraw and deposit) via multiple persons (you and your father) on a joint account object. Each person will run via separate thread and performs the banking operations.

For withdrawing money, at first check the available balance, then if sufficient balance is there, then deduct the money and display the current thread, withdrawn amount and current balance. For depositing money, after adding the money to account show the name of the current thread, deposited amount, current balance. Assume you and your father are trying to withdraw/deposit at same time from different ATM then at a time only one person should be allowed to update the balance. Set the name of the threads as "Son" and "Father".

Test the following case: Initial balance 500. Son withdraws 500 and sleep for 5 secs. Father tries to with draw 500 and get Not sufficient balance message. Then he deposits 10000. Son wakes up and with draw 1000 more.

```
>Son: Withdrawn = 500.0, Current balance = 0.0
>Father: Not sufficient balance
>Father: Deposited = 10000.0, Current balance = 10000.0
>Son: Withdrawn = 1000.0, Current balance = 9000.0
```

3.

Write a program to take input set of pairs {movie name, director name} from command line and store all the strings in an array. Perform the following operations on all the strings:

a. Covert all the strings in upper case
b. Print all the movies starts with a given character
c. Print the count of the no of movies by different directors.

Test the following case:

```
>java Movie “Bommarillu,bhaskar”  “Eega,rajamouli” “manam,vikram” “Bahubali,rajamouli”
> Enter a character: B
> BOMMARILLU,BHASKAR
> BAHUBALI,RAJAMOULI
> Count of movies by different directors:
> RAJAMOULI – 2
> BHASKAR – 1
> VIKRAM - 1
```

4.

Write a class Bicycle with two instance variables gear and speed and two constructors, default and parametrized. The default constructor should call the parametrized constructor and pass the default values of gear = 1 and speed = 0. Add methods applyBrake(), speedup() and toString() method to print info of Bicycle. The speedup() should check that max speed is not more than 25 km/hr.

Create a sub class MountainBike with extra parameter seatHeight with default gear 4 and seat height is 5. Add two constructors, default and parametrized and two methods setSeatHeight() and toString(). Max speed of mountain bike is 35 km/hr.

Create person class with a drive method that receives Bicycle or MountainBike object. The drive method should call all the methods based on whether the person is driving Bicycle or MountainBike.

Write a Test class and call the drive() method of Person class to get the following output:

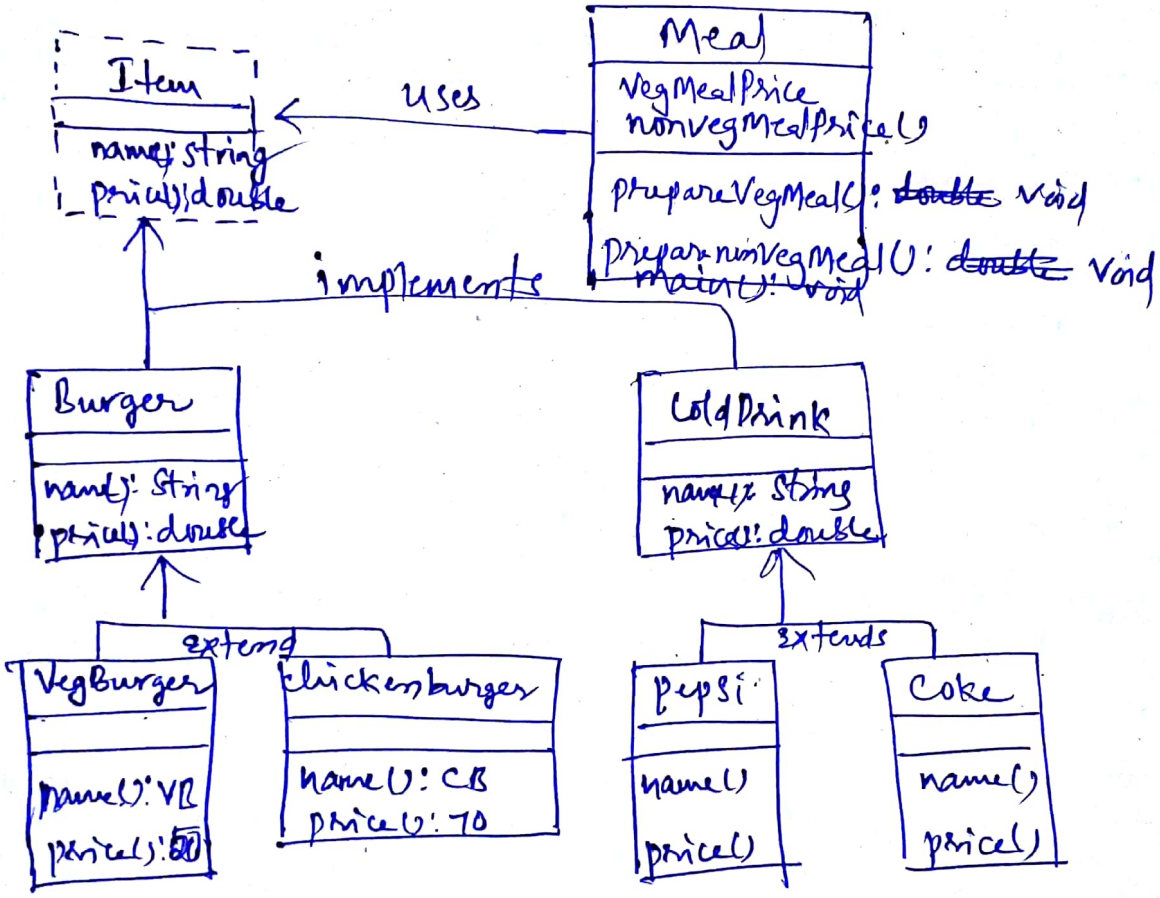
```
>Driving Bicycle
>No of gears are 1, Speed is 0 km/hr
>Speed up by 20 km/hr
>No of gears are 1, Speed is 20 km/hr
>Applying brake: speed reduced by 20 km/hr
>Got down from Bicycle
>Driving Mountain Bike
>No of gears are 4, Seat Height 5-inch, Speed is 0 km/hr
```

```
>Adjusting seat height: Seat height set to 10 inches
>Speed up by 40 km/hr
>Max speed cannot increase 35 km/hr
>Speed up by 40 km/hr
>No of gears are 4, Seat Height 10-inch, Speed is 30 km/hr
>Applying brake: speed reduced by 30 km/hr
```

Section 2(1x16=16 M)

1. Considered a business case of fast-food restaurant where a typical meal could be a burger and a cold drink. Burger could be either a Veg Burger or Chicken Burger. Cold drink could be either a coke or pepsi. We are going to create an *Item* interface representing food items such as burgers and cold drinks with two methods name() (name of item) and price() (price of item). Two abstract Classes *Burger* and *ColdDrink* implement *Item* Interface. Concreate classes vegBurger and ChikenBurger are subclasses of Burger, Concreate classes Coke and Pepsi are subclasses of *ColdDrink.* All the above classes should be present in pakage items.

Create Meal Class with methods prepareVegMeal()(should return veg Meal Price), prepareNonVegMeal(should return non veg Meal Price), here veg meal is combination of veg burger and coke, non veg meal is combination of chickenburger and Pepsi. Take input from user for type of meal and print price of the meal. If user entered meal type is not avilable throw custom exception with message "that type of meal is not avilable"


Item
name(): String
price(): double
uses
Meal
VegMealPrice
nonvegMealPrice()
prepareVegMeal(): ~~double~~ void
prepareNonVegMeal(): ~~double~~ void
main(): void
implements
Burger
name(): String
price(): double
ColdDrink
name(): String
price(): double
extend
extends
VegBurger
name(): VB
price(): 80
ChickenBurger
name(): CB
price(): 70
Pepsi
name()
price()
Coke
name()
price()