National University of Computer and Emerging Sciences, Lahore Campus

Course:	Computer Programming	Course Code:	CS103
Program:	BS(Computer Science)	Semester:	Fall 2018
Duration:	135 Minutes	Total Marks:	90
Paper Date:	21-Dec-2018	Weight	30
Section:	All	Page(s):	3
Exam:	Final – Part II	Roll No:	
		Section	

Instructions: - Questions during exam are not allowed. Take reasonable assumptions where needed.

Question 1[30 Marks]: We are developing an application for a pick and drop service "**PnD**". PnD deals with the Customers, Drivers and Ride and its Object Oriented Design is as follows:
A **Customer** and a **Driver** is a **Person** having a *PhoneNo* for registration and a *Name*.
A **Driver** has a **Vehicle**.
A **Vehicle** has a *Category (*1 = Bike, 2 = Rikshaw, 3 = Car), *color* and *registrationNo*.
A **Customer** has accountBalance/wallet information.
A **Ride** has a *StartTime*, *EndTime*, *StartLocation*, *EndLocation,* reference of driver*, reference of Customer, RideDuration, RideDistance and Cost.*
Both the Driver and Customer have RideHistory.
Definitions of all the classes used in this application are given below. **Your task is to write Destructors of all the classes being used in this Application.**

```
class DateTime{

// Suppose this class provides full date and time functionality and does not require destructor

};
```

```
Class Location{

//Suppose this class provides full functionality of location coordinates and does not require destructor

};
```

```
Class Person{

Private:

        Char* phoneNo; //required for Registration

   Char* name;

        Ride** ridesHistory; //Containing references of all

                                  //the rides of a person
```

```
Class Vehicle{

Private:

    int category;    // 1= Bike, 2= Rikshaw,
3=Car

    Char* color;    // "Black", "White" etc.

    Char* registrationNo;   //e.g. "LEC-1234"

};
```

```
Class Ride{

Private:

    DateTime startTime;

    DateTime endTime;

    Location stratLocation;

    Location endLocation;

    Driver* driver;     // reference of driver of
ride

    Customer* customer;      //reference of
customer of

                                        //ride

    Float rideDuration;          //(endTime -
startTime)

    Float rideDistance;        //Total distance
covered

    Float cost;                  //Cost of the ride

Public:

    Static int totalRides;

};
```

```
};
```

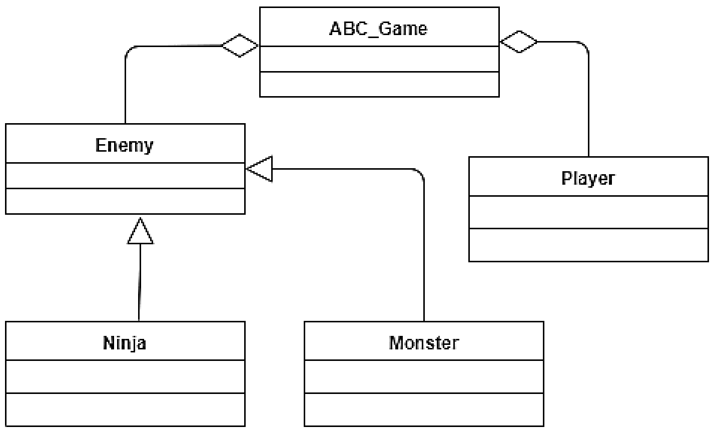
```
Class Customer : public Person{

Private:

    int accountBalance;

//For Example, accountBalance = -50
means //customer needs to pay Rs. 50 to
PnD and vice versa //for accountBalance =
50

Public:

    Static int totalCustomers;

};
```

```
Class Driver : public Person{

Private:

    Vehicle vehicle;   //Information about
Driver's

                            //vehicle

Public:

    Static int totalDrivers;

};
```

```
Class PnD{

Private:

    Customer** allCustomers; //list of all
Customers

    Drivers** allDrivers;              //list of all
Drivers

    Ride** allRides;                   //list of all
Rides

};
```

Question 2[30 Marks]: We have to write C++ implementation of a game "ABC_Game".
The **Game** has one player and enemies (0 to 10).
A **Player** has (x, y) coordinates on screen and a life limit.
An **Enemy** has some (x, y) coordinates on screen and it can Attack the Player.

A **Monster** is an Enemy. If its distance from player is **more than 10 units**, he can attack the player with Gun. After the attack of Monster, the player's life will reduce by 2 and a message "Monster attacked with a Gun" will be displayed.
A **Ninja** is also an Enemy. If its distance from player is **less than 10 units**, he can attack the player with knife. After the attack of Ninja, the player's life will reduce by 1 and a message "Ninja attacked with a knife" will be displayed. Consider the given UML diagram. Where empty diamonds and triangles represent aggregation and inheritance, respectively. **Your task is to write a C++ program using concepts of Object Oriented Programming, while satisfying the above description of classes. Your program should run successfully for the driver program given below without any memory leakage, runtime exceptions and extra memory consumption.**



```
void main()

{

    ABC_Game game;    /*Game can have only one player and zero to 10 enemies.
Initially there is no player and no enemy*/



    int x = GetRandomScreenCoordinates_X();

    int y = GetRandomScreenCoordinates_Y();   /* Suppose these functions are
already implemented and give random screen coordinates. You do not need to
write GetRandomScreenCoordinates_X() and GetRandomScreenCoordinates_Y() */



    game.AddPlayer(new Player(x, y, 10));   /*Adds new player in game. Sets its
coordinates to x and y and its life limit to 10*/



    while(game.GetPlayersLifeCount()>0)

    { //this loop will run until the player's life becomes zero
```

```
        int randomNumber = rand()%10;   //generate a random number between
1 to 9

        if(randomNumber == 1)

        {

            int x1 = GetRandomScreenCoordinates_X();

            int y1 = GetRandomScreenCoordinates_Y();

            game.AddEnemy(new Ninja(x1, y1)); /*Adds a Ninja enemy with
coordinates x1, y1 in array of Enemies in game*/

        }

        else if(random Number==2)

        {

            int x1 = GetRandomScreenCoordinates_X();

            int y1 = GetRandomScreenCoordinates_Y();

            game.AddEnemy(new Monster(x1,y1));/*Adds a Monster enemy with
coordinates x1, y1 in array of Enemies in game*/

        }

        game.Play();    /*this function will call the attack function of all the enemies
currently present in game*/



        game.MovePlayer();    /*Suppose this function randomly updates the
coordinates and moves the player. You do not need to write this function*/

    }//End of While

}//End of Main
```

Note: Suppose you are already given a global function to compute distance between two coordinates. You do not need to write it.
int CalculateDistance(int x1, int y1, int x2, int y2);   //Calculates distance between (x1,y1) and (x2,y2)

Question 3[30 Marks]:

A rare book collector recently discovered a book written in an unfamiliar language that used the same characters as the English language. The book contained a short index, but the ordering of the items in the index was different from what one would expect if the characters were ordered the same way as in the English alphabet. The collector tried to use the index to determine the ordering of characters (i.e. the collating sequence) of the strange alphabet, then gave up with frustration at the tedium of the task. You are to write a function **Char* FindCharactersSequence(char** listOfStrings)** to complete the collector's work. Your program will take an array of strings that has been sorted according to a particular collating sequence and determine what that sequence is.

Sample Input:

```
XWY
ZX
ZXY
ZXW
YWWX
```

Output:

Correct output for the input data above is:

```
XZYW
```

(Note: The collating sequence is the order in which characters are sorted.)

Solution Question 1:

```
Vehicle::~Vehicle()
{
        If (color != 0) delete[] color;
        If (registrationNo != 0) delete[] registrationNo;
}
Virtual Person::~Person()
{
        If (phoneNo != 0) delete[] phoneNo;
        If (name != 0) delete[] name;
        If (ridesHistory != 0) delete[] ridesHistory;
}
Customer::~Customer(){//Does Nothing}
Driver::~Driver(){//Does Nothing}
Ride::~Ride(){//Does Nothing}
PnD::~PnD()
{
        If(allCustomers)
        {
                For(int i=0; i< Customer::totalCustomers; i++)
                        Delete allCustomers[i]
                Delete[] allCustomers;
        }
        If(allDrivers)
        {
                For(int i=0; i< Driver::totalDrivers; i++)
                Delete allDrivers[i]
            Delete[] allDrivers;
        }
        If(allRides)
        {
            For(int i=0; i< Ride::totalRiders; i++)
                Delete allRiders[i]
            Delete[] allRiders;
        }
}
```