

Object Oriented Programming-CS217

Name: Muhammad Istafa Malik

Roll: P19-0033

Section: BS(CS)19-2A

Q1:

Q1

①

- M. IstaFa Malik

include <iostream>

include <string>

using name space std;

class Author

{

string name;

string email;

char gender;

Public:

Author (string name, string email, char gender)

{

this → name = name;

this → email = email;

this → gender = gender;

}

string getName()

{

return name;

}

string getEmail()

{

return email;

}

(2)

void setEmail(string email)

{

this->email = email;

}

char getGender()

{

return gender;

}

string toString()

{

return ("Author[name = " + name + "email = " +
email + "gender = " + gender);

}

} ;

class Book : public Author {

private :

String name;

Author author;

double price;

int qty = 0;

public :

Book(String name, Author author, double price)

{

this->name = name;

this->author = author;

this->price = price;

}

Book(String name, Author author, double price,

{

int qty)

this->name = name;

this->author = author;

this->price = price;

this->qty = qty;

}

(5)

string getName()

{

 return name;

}

string Author getAuthor()

{

 return author;

}

double getPrice()

{

 return price;

}

void setPrice(double price)

{

 this → price = price;

}

int getQty()

{

 return qty;

}

void setQty(int qty)

{

 this → qty = qty;

}

(3)

String toString()

{

return Book[name = " + name + toString() +
" price = " + price + " qty = " + qty);

}

(4)

String getAuthorName()

{

return getName();

}

String getAuthorEmail()

{

return getEmail();

}

char getAuthorGender()

{

return getGender();

}

};

⑥

```
int main()
```

```
{  
    Author A("A4", "ali@gmail.com", M);
```

```
    Book b1("A4",  
            b1("C++", A, 235.75, S));
```

```
    cout << "Name of Author" << b1.getAuthorName();
```

```
    cout << "Email of Author" << b1.getAuthorEmail();
```

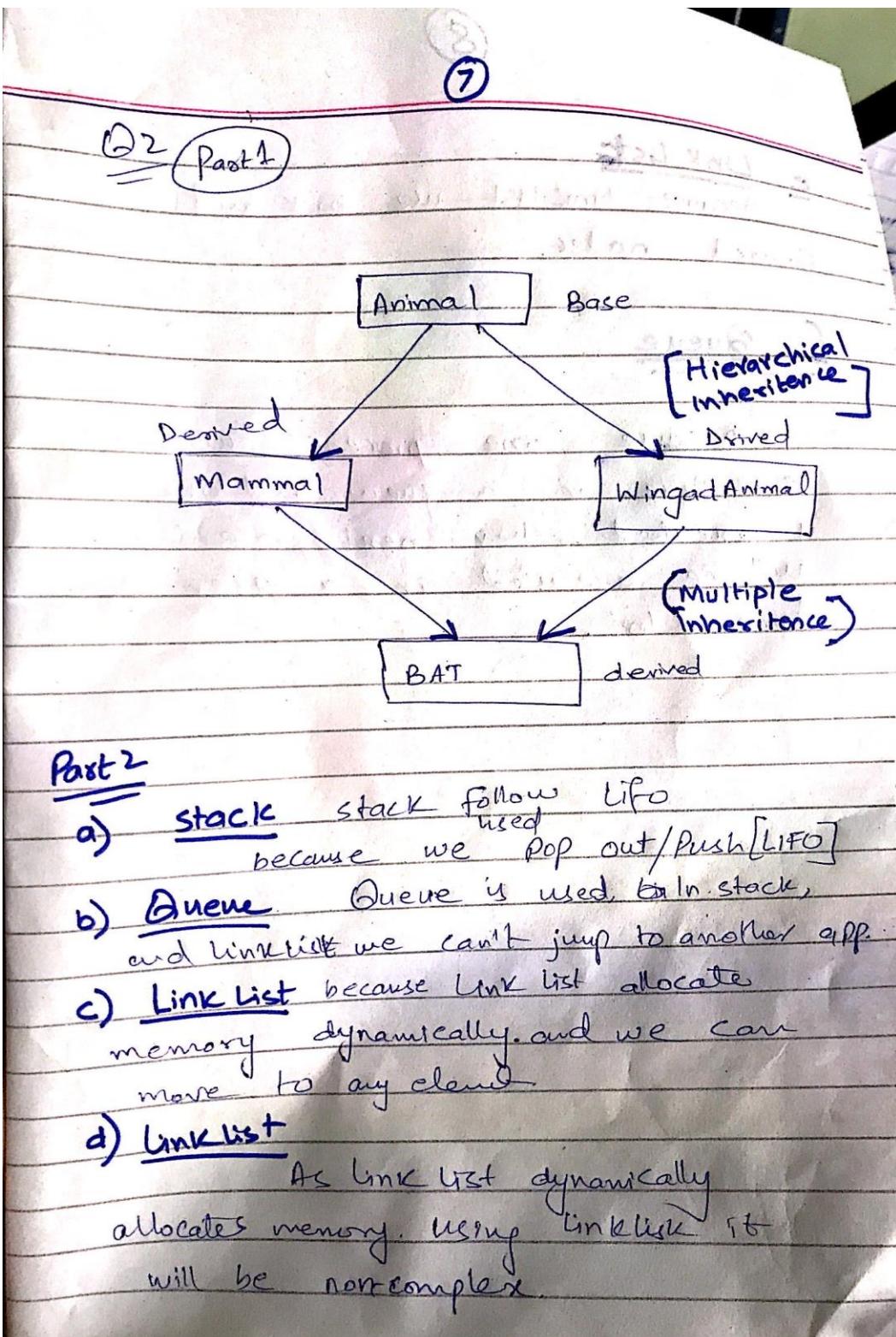
```
    cout << "Gender of Author" << b1.getAuthorGender();
```

A:

```
return 0;
```

```
}
```

Q2:



⑧

e Link List

Because Link List use back and forward nodes.

f Queue

because one process will be stored in a queue and one is executed, then next process will be executed in a given time slot.

Q3:

⑨

Q3

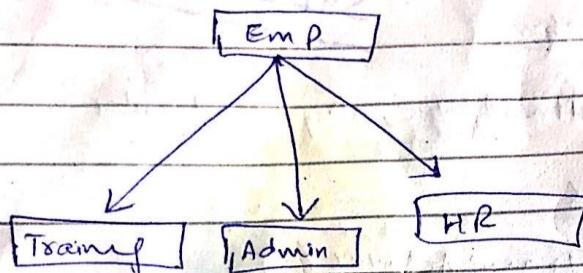
classes

i) Emp Base class

ii) Training derived class from Emp

iii) Admin " " " "

iv) HR " " " "



Data members & Methods

Emp

Empnumber, EmpName, EmpJob, ~~EmpType~~

EmpDoj, EmpSal, cityName, countryName

villapincode, Pincode number,

Abstract class

Methods

Emp (Empnumber, EmpName, EmpJob, EmpDoj,

EmpSal, cityName, countryName, villageName -

PincodeNumber) { }

setEmpData();

getEmpData();

virtual setEmpRegistration(); → Abstract method

(10)

virtual DeparmentActives();

virtual TransActives();

virtual Report();

Abstract method

Abstract method

Abstract method

?;

Trans

Data member

DepartmentType, DepartmentNumber

Subclass of Emp

Method

TransActives();

Report();

HR

Data member

DepartmentType, DepartmentNumber

Subclass of Emp

methods

Registrable^{EP} modul();

DepartmentActv();

Report();

Admin

Data member
DepartmentType; Department;

methods

RegistrableDep();
Depachetib();
Report();

3 phenomenon

It is Inheritance,

~~multiple~~ b/c subclasses inherit the
base class iEmp

Q3

#include <iostream>

!!

y) #include <smif>
using namespace std;

class Emp {

protected:

int Empnumber;

String EmpName;

String EmpJob;

String EmpDoj;

double EmpSal;

String CityName;

String CountryName;

String VillageName;

double pincodeNumber;

PUBLIC:

Emp (Empnumber, EmpName, Emp

Emp (int Empnumber, Emp { });

void EmpData (int Empnumber, String EmpName;

String EmpDoj, double EmpSal; String CityName;

String CountryName; String VillageName; double

PincodeNumber)

{

this->Empnumber = Empnumber;

this->EmpName = EmpName;

this->EmpDoj = EmpDoj.

(11)

(12)

this → Eipsal = Eipsal;

this → cityname = cityname;

this → countryname = countryname;

this → villageName = villageName;

this → pincodeNumber = villagePincodeNumber;

}

void getEipsal();

{

cout << Eipsal;

cout << Eipsal;

cout << EipName;

cout << EipNo;

cout << EipSal;

cout << CityName;

cout << CountryName;

cout << VillageName;

cout << PincodeNumber;

}

virtual ~~set~~ Registration();

virtual DepartActivities();

virtual TrainActivities();

virtual Report();

{

(13)

Class Training : Public Emp

{

string DepartmentType;

double DepartmentNumber;

public:

void TrainActivities()

{

cout << "Train activities";

}

void Report();

{

cout << "Training Report";

}

};

Class Admin : Public Emp

{

string DepartmentType;

double DepartmentNumber;

public:

void RegisterEmp();

{

cout << "Register Emp";

}

(14)

void DepartmentActions()

{

cout << "Department Action By Admin";

}

void Report();

{

cout << "Admin Depart Report";

{

};

class HR : public Emp

{

string DepartType;

double DepartNumber;

public

{

void EmpRegistration();

{

cout << "Emp Registration by HR";

{

void DepartmentActions()

{

cout << "Department Actions By HR";

{

(15)

void Report()

{

* cout << "Report File";

}

};

int main()

{

echo;

}

Q4:

Q.4 Part 1

16

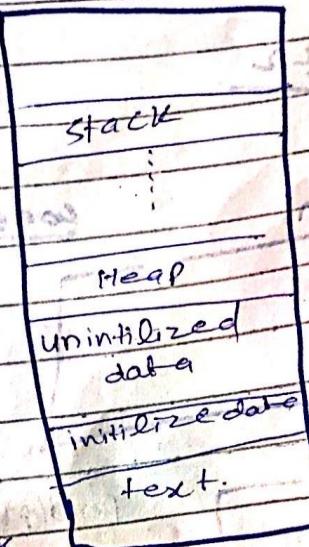
High address

using namespace std;

```
int x=0;
static int count=1, z;
const char a='A';
```

global variables stored
in initialized Data
segment.

Low Address



```
int main()
```

```
{ int s=new int;
```

```
int y=10;
```

const char b="B";

```
static int c=6
```

In uninitialized variable
segment

static variable
stored in initialized
Data segment

```
return 0;
```

```
}
```

17

Q4
Part 2

