Sunny kumar

batch-Linux System Programming Track

Q. **pass by refrence**

```
#include<iostream> //header file

using namespace std;

class point{ // creating a class name as a point

public: //make it public accessifier

double x,y; //variable defination

};

void ofsetpoint(point &p,double x, double y) //function

{

p.x +=x;

p.y +=y;
cout<<"inside address of p"<<&p<<endl;

}

int main()

{

point p; //create an object of a class

p.x=3.0;

p.y=4.0;

ofsetpoint(p,1.0,2.0); // funtion called

cout << "(" << p.x << " , " << p.y << ")";

cout<<"outside address of p"<<&p<<endl;

}
```

Q. default constructor program

```
#include<iostream> //header file

using namespace std;

class constructor{ // creating a class name as a constructor

public: //make it public accessifier

int a,b; //define ttwo integer variable a nd b

constructor() //default constructor with no parameter
{
a=10; //assign the value of a
b=20; //assign the value of b
}
};
int main() //here is the main function
{
constructor dc; //create an object of a class
cout<< "a is :"<<dc.a<<endl<<"b is :"<<dc.b<<endl; // print the value of a ans b
return 0;
}
```

Q. parameter constructor program

```
#include<iostream>
using namespace std;
class point{
public: //make it public accessifier
double x,y;
point(double n,double m)

{

x=n;

y=m;

cout<<"parametr constructor called"<<endl<<"x is"<<x<<"y is"<<y<<endl;

}

};

int main()

{

point p(1,2); //create an object of a class and assing value

cout<<"x is"<<p.x<<" "<<"y is"<<p.y<<endl;

cout<<"constructor called"<<endl;

p.x=3.0;

p.y=4.0;

cout << "(" << p.x << " , " << p.y << ")"; //print x and y value

cout<<"outside address of p"<<&p<<endl;

}
```

Q. inheritance program

1.

```
#include<iostream>

using namespace std;

class account{

public: //make it public accessifier

float salary=13000;

};

class emp:public account{

public: //make it public accessifier

float bonus=3000;

};

int main()

{

emp E1; //create an object of a class

cout<<"salary is :"<<E1.salary<<endl;

cout<<"bonus is :"<<E1.bonus<<endl;

return 0;

}
```

2.

```
#include <iostream>

using namespace std;

class Animal {

public: //make it public accessifier

void eat() {

cout << "This animal is eating." << endl; ////print the statement

}

};

class Dog : public Animal {

public: //make it public accessifier

void bark() {

cout << "The dog is barking." << endl; //print the statement

}

};

int main() {

Dog myDog; //create an object of a class

myDog.eat(); // Inherited method
myDog.bark(); // Child class method
return 0;
}
```

Q. Encapsulation program

```
#include <iostream>
using namespace std;
class rectangle {
private: //make it private accessifier
double length;
double width;
public: //make it public accessifier
void setLength(double l) { // Setter methods
length = l;
}
void setWidth(double w) {
width = w;
}
double getLength() const { // Getter methods
return length;

}

double getWidth() const {

return width;

}

double calculateArea() const {

return length * width;

}

};

int main() {

rectangle R; //create an object of a class

R.setLength(5.0); //set length and width

R.setWidth(3.0);

cout << "Length is " << R.getLength() << endl; //print length

cout << "Width is: " << R.getWidth() << endl; //print width

cout << "Area of rectangle is: " << R.calculateArea() << endl; //print area

return 0;

}
```