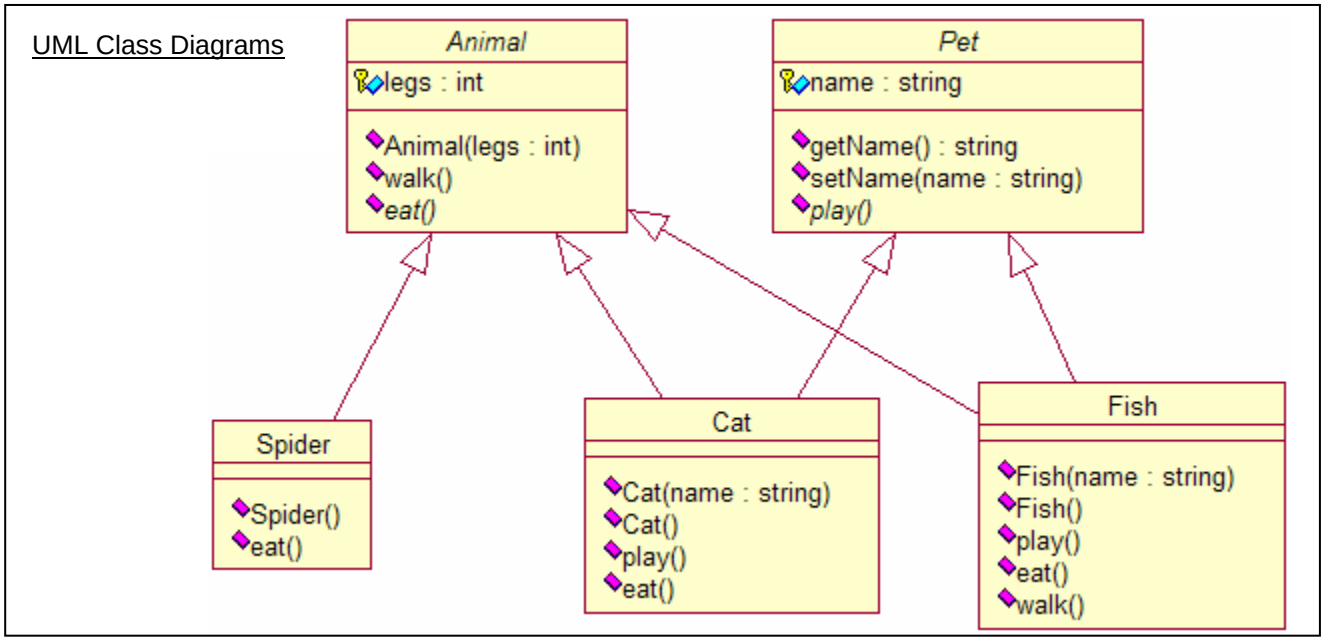
OBJECT ORIENTED PROGRAMMING

2nd Midterm Exam

(There are 3 Questions. 2-Hour Exam)

Q.1) You will create a hierarchy of animals that is rooted in an abstract class Animal. Several of the animal classes will implement an abstract class called Pet. UML class diagram is given in the following diagram.



(a) (10) Write the class definitions.

(b) (30) Give the implementations of the methods considering that the application code produces the output given inside the box.

```
int main(int argc,char *argv[]){
     Fish *f = new Fish("Jaws");
     Cat *c = new Cat("Tekir");
     Animal *a = new Fish();
     Animal *e = new Spider();
     Pet *p = new Cat();

     f->play();
     c->play();
     e->eat();
     e->walk();
     a->walk();
     p->play();
     return 0;
}
```

application code

output

```
Jaws is playing now.
Tekir is playing now.
Spider is eating now.
Animal with 8 legs is walking.
Fish cannot walk.
Garfield is playing now.
```

Q.2) (30) What is the output of the following C++ code?

```
#include <iostream>
using namespace std;
class Base {
    public:
         int hold;
         Base(int hold=0){this->hold=hold;}
         virtual void f(){
             hold=2;
             cout << endl << "hold: " << hold ;
         }
         virtual void g()=0;
         void h(){
             hold=4;
             cout << endl << "hold: " << hold ;
         }
} ;
class Derived : public Base {
    public:
         int hold;
         Derived(int hold1=0,int hold2=0):Base(hold2){
             this->hold=hold1;
         }
         void f(){
             hold=8;
             cout << endl << "hold: " << hold ;
         }
         void g(){
             hold=16;
             cout << endl << "hold: " << hold ;
         }
         void h(){
             hold=32;
             cout << endl << "hold: " << hold ;
         }
} ;
int main(int argc,char *argv[]){
      Base *bp = new Derived(-2,-4) ;
    ① cout << endl << bp->hold;
    ② bp->f();
    ③ bp->g();
    ④ bp->h();
      Derived *dp= static_cast<Derived*>(bp);
    ⑤ cout << endl << dp->hold;
    ⑥ dp->f();
    ⑦ dp->g();
    ⑧ dp->h();
      return 0;
}
```

Q.3) (30) What is the output of the following C++ code?

```
class B {
  public:
     B(){ cout << "\nB's constructor"; }
     ~B(){ cout << "\nB's destructor"; }
};
class C {
  private:
     static int n;
  public:
     C(){ cout << "\nC's constructor # " << ++n ; }
     ~C(){ cout << "\nC's destructor # " << n-- ; }
};
int C::n=0;
class A {
  protected:
     B *b;
     C c;
  public:
     A(){  b= new B();
           cout << "\nA's constructor"; }
     ~A(){ delete b;
           cout << "\nA's destructor";  }
};
class D : public A {
  protected:
     B b;
     C c;
  public:
     D(){ cout << "\nD's constructor"; }
     ~D(){ cout << "\nD's destructor"; }
};
int main(){
    D a;
    return 0;
}
```