



100
100
Amazing

Jordan University of Science & Technology
Department of Computer Science
CS 112 – Exam #2 (29/4/2010) -- Form D

Name

#:

Section #: 9

Day & Time: Thursday

Instructor: *Ali Alkhalaf*

Answer all questions as indicated. Closed book/Closed Notes.
NO PDAs (calculators, handheld devices, cell phones, etc.) allowed.

Part1: Basic Concepts (15 points)

Q1) Select the correct answer for all of the following 5 questions, (3 Points each)

1.1) D classes are new classes that are created from existing classes.

A: Base

B: Composite

C: Concrete

D: Derived

1.2) B refers to the ability to combine data, and the operations on that data, in a single unit.

A: Inheritance

B: Encapsulation

C: Composition

D: Polymorphism

1.3) C is the ability to use the same expression to denote different operations.

A: Inheritance

B: Composition

C: Polymorphism

D: Encapsulation

1.4) B takes place when one or more members of a class are objects of another class type.

A: Inheritance

B: Composition

C: Polymorphism

D: Protection

1.5) Redefining a member function of a base class is also known as A the function.

A: overriding

B: overloading

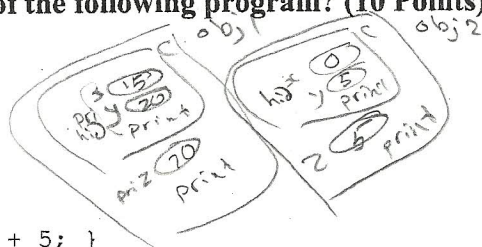
C: protecting

D: specifying

Part 2: Program Analysis (30 Points)

2.1) What is the exact output of the following program? (10 Points)

```
#include<iostream>
using namespace std;
class C{
    int x, y;
public:
    C(){ x = 0; y = 5; }
    C(int i){ x = i; y = x + 5; }
    void print(){
        cout << x << endl << y << endl << "C" << endl;
    }
};
class D: public C{
    int z;
public:
    D(){ z = 5; }
    D(int i): C(i + 15) { z = i + 20; }
    void print(){
        C::print();
        cout << z << endl; cout << "D" << endl;
    }
};
void main()
{
    D obj1(0), obj2;
    obj1.print();
    obj2.print();
}
```

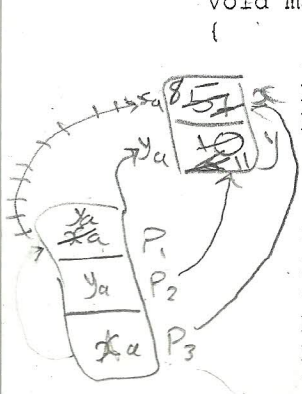


Output: of 2.1

15	/
20	/
C	/
20	/
D	/
0	/
5	/
C	/
5	/
D	/

2.2) What is the exact output of the following program? (10 Points)

```
#include<iostream>
using namespace std;
void main()
{
    int x=5, y=10, *p1, *p2, *p3;
    p1 = &x;    p2 = &y;
    p3 = new int;
    *p3 = 15;
    *p2 = *p1 + 1;
    *p1 = *p1 + 2;
    x++;
    y = y + *p3 - 10;
    delete p3;
    p3 = p1;
    p1 = p2;
    cout << x << endl << y << endl
        << *p1 << endl << *p2 << endl
        << *p3 << endl;
}
```



$p_1 = x = 7$
 $p_2 = y = 11$
 $p_3 = 15$
 $p_1 = p_2 = y = 11$

Output: of 2.2

7	/
11	/
11	/
11	/
15	/

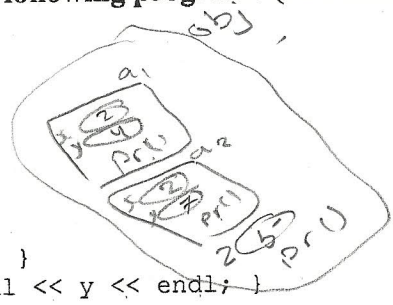
2.3) What is the exact output of the following program? (10 Points)

```
#include<iostream>
using namespace std;

class A{
    int x, y;
public:
    A(){ x = 2; y = 4; }
    A(int i){ x = i; y = x + 5; }
    void pr(){ cout << x << endl << y << endl; }
};

class B{
    int z;
    A a1, a2;
public:
    B(): a2(2) { z = 5; pr(); }
    B(int i): a2(5) { z = i + 10; }
    void pr(){ a1.pr(); a2.pr(); cout << z << endl; }
};

void main()
{
    B obj;
}
```



Output:

2	✓
4	✓
2	✓
7	✓
5	✓

Part3: Programming (55 points)

Problem Description: consider a class named **Person** and another named **Student**. The **Person** class has 2 protected member variables: **Name** and **ID** and a default constructor that sets **Name** to "???" and **ID** to -1. The **Student** class publically inherits the properties of the **Person** class and adds 3 private member variables: **Number_of_Taken_Courses**, the array **Marks[50]**, and **Department Name**. The array **Marks** keeps track of student marks for all taken courses. The **Student** class has 5 public member functions:

1. Default constructor to initialize the **Name** to empty, **ID** to zero, **Department_Name** to empty, **Number_of_Taken_Courses** to zero and all elements in the array **Marks** to -1.0.
2. Function **setStudentInfo** to set the values of the student (**Name**, **ID**, **Department_Name**)
3. Function **printStudentInfo** to print the **Name** and **ID** and **Department_Name** in the following format: **Name *** ID *** Department_Name *** GPA**
4. Function **insertMark** to add one mark to the array **Marks**.
5. Function **getGPA** that returns the **GPA** (the average of all student marks)

Questions:

3.1) Write a complete definition of the **Person** class, do not include the implementation for member functions, just their prototype (5 Points).

- 3.2) For the **Person** class, write the implementation of its default **constructor**. (3 points)
- 3.3) Write a complete definition of the **Student** class, *do not include the implementation for member functions, just their prototype*. (14 Points)
- 3.4) For the **Student** class, write the implementation of its default **constructor**. (6 points) (1)
- 3.5) Write the implementation of the member function **insertMark**. (6 Points) (4)
- 3.6) Write the implementation of the member function **getGPA**. (10 Points) (1)
- 3.7) In the **main** function:
- 3.7.1) Declare a pointer named **stPtr** of type **Student**. (2 Points)
- 3.7.2) Dynamically allocate an object of type **Student** and save its base address in **stPtr** (2 Points)
- 3.7.3) Set values to member variables of the object pointed to by **stPtr** as the following ("Ali", 99, "CS"). (2 Points)
- 3.7.4) Insert the first mark **90** to the object pointed to by **stPtr**. (2 Point) → *for insert mark part 1 of 3*
- 3.7.5) Insert the second mark **95** to the object pointed to by **stPtr**. (1 Point)
- 3.7.6) Print the values of the object pointed to by **stPtr**. (2 Points)

Solution r

(3.1)

```
class Person {
protected:
```

```
    String Name;
```

```
    double ID;
```

```
public:
```

```
    person();
```

// this is the constructor of the class.

// Post conditions: ~~give~~ initialize the variables

Name = "???" and

```
    // ID = -1
```

```
};
```

(3.2)

```
person::person() {
```

```
    Name = "??";
```

```
    ID = -1;
```

```
}
```

(3,3)

```

class student : public Person {
private :
    int NOofCourses ; // holds the number of Taken Courses
    double Marks [50]; // holds the Marks of the students
    string DepartmentName ; // holds the department name.

public :
    student () ; // a constructor to initialize Name and Department
    void setStudentInfo (string N, double I, string DN) ;
    void printStudentInfo () ;
    void InsertMark (double M) ;
    double getGPA () ;
};
    
```

(3,4)

```

student :: student () {
    Name = "" ;
    ID = 0 ;
    DName = "" ;
    NOofCourses = 0 ;
    for (int i=0; i<50; i++)
        Mark[i] = -1 ;
}
    
```

(3,5)

```
void student :: InsertMark (double M) {  
    for (int i=0; i<50; i++)  
        if (Marks[i] != -1) {  
            Marks[i] = M; i=50; }  
}
```

(3,6)

```
double student :: getGPA () {  
    double x=0, y=0, GPA;  
    for (int i=0; Marks[i] != -1; i++) {  
        x += Marks[i];  
        y++; }  
    GPA = x / y;  
    return GPA;  
}
```

(3,7)

```
void main () {  
    student *stPtr;  
    stPtr = new student;  
    stPtr -> setStudentInfo ("Ali", 99, "CS");  
    stPtr -> InsertMark (90);  
    stPtr -> InsertMark (95);  
    stPtr -> PrintStudentInfo ();  
} // end main.
```