National University of Computer & Emerging Sciences
FAST-Karachi Campus
CS201- Data Structures (Fall 2018)
Quiz#1

Dated: September 11, 2018 Marks: 30
Time: 20 min. Std-ID: _____Sol_____

Question No. 1 Appropriately match the columns. [10]

Column A	Column B
1. Object (b)	a. Virtual Base Class
2. Function Overriding (e)	b. Memory + Operations
3. Virtual Destructor (j)	c. Constant Pointer
4. RTTI (d)	d. TypeID()
5. Reference (c)	e. Run Time Polymorphism
6. Compile Time Polymorphism (g)	f. Operators preference + association
7. Pure virtual function (i)	g. Function Overloading
8. Expression (f)	h. Class
9. Data Type (h)	i. Abstract base class
10. Derived + One Base memory (a)	j. Base Class Destructor

Question No. 2 Indicate TRUE or FALSE and explain in 2-3 lines to the argument on it.

a. A derived class pointer can hold a base class object.

FALSE. A derived class pointer is not directly allowed to hold a base class object. C++ is a statically typed language, and allowing implicit Base-to-Derived conversions and a pointer to Base class can hold any Derived class object hence it is a legitimate but it do not allow the vice versa.

b. A virtual or pure virtual function can be private.

TRUE, when you need to make specific behavior in a base class customizable in derived classes, while protecting the semantics of the interface (and/or the base algorithm therein), which is defined in public methods that call private virtual methods.

c. A destructor can be virtual? Give an example

TRUE. It is common to have a virtual destructor when you have dynamic memory in your base class and derived class. You need to wipe-off memory from the base class as well. Making a destructor take case of these things.

Question No. 2 Given this skeleton class and partial implementation; provide the implementation for the commented functions with question marks (?).

```
class Date{

   private:

 int *DateData; //day/month/year

  public:

  // default constructor

  Date() {

   DateData= new int[3];

     *(DateData+0)=0;

     *(DateData+1)=0;

    *(DateData+2)=0; }

  Date(const int a[]) {

   DateData= new int [3]   ;
for( int i=0; i < 3; i++)

  { *(DateData+i)=*(a+i);

  }  }

  Date(const Date & rhs) {

   DateData= new int [3]   ;

  for( int i=0; i < 3; i++)

   { *(DateData+i)=
*(rhs.DateData+i); }  }

 ~ Date() {

     if(DateData != 0) {

    delete [] DateData; }  }

};  // class end
```

```
int main(){

Date D1, D2;

int a[3]= {10, 10, 1997};

cout << D1.getDay()  << D1.getMonth()
<< D1.getYear() << endl;

Date D3(a);

cout << D3.getDay()  << D3.getMonth()
<< D3.getYear() << endl;

return 0;


}
```