UNIVERSITY OF CENTRAL PUNJAB, LAHORE

Mid-Term Solution Fall 2022

Course Title: Object Oriented Programming **Time:** 90 minutes
Course Instructor: _____ **Program Name:** BS(SE)
Section: _____ **Department:** Software Engineering

Total marks: 50 marks **Obtained Marks: _____**

Question # 1: Please, answer the following short questions. [15 Marks]

Part (a). What are the two cases when C++ performs member wise (bitwise) assignment on objects.[02Marks]

Copy Constructor [1]
Assignment Operator [1]

Part (b). What is "this" pointer and its type? Mention at least two scenarios when we use "this" pointer? [03 Marks]

The 'this' pointer is passed as a hidden argument to all non-static member function calls and is available as a local variable within the body of all non-static functions. [1]

1) **When local variable's name is same as member's name [1]**
2) **To return reference to the calling object [1]**

Part (c). What is the purpose of copy constructor? Write down the conditions when copy constructor is called by the compiler? [03 Marks]

The copy constructor is used to copy data of one object to another. [1]

1) **When an object of the class is returned by value. [0.5]**
2) **When an object of the class is passed (to a function) by value as an argument. [0.5]**
3) **When an object is constructed based on another object of the same class. [0.5]**
4) **When the compiler generates a temporary object. [0.5]**

Part (d). Suppose we create a class with empty body (i.e it has no data member and member functions) and then we compile it. Please name the functions that this class will have once compiled. [02 Marks]

1) **Constructor [1]**
2) **Destructor [1]**

Part (e). What are few of the restrictions while overloading the operators? [03 Marks]

1) **Only built-in operators can be overloaded. [1]**
2) **Arity of the operators cannot be changed. [1]**
3) **Precedence and associativity of the operators cannot be changed. [1]**

Part (f). Please name, what are four pillars of OOP? Define any one of them. [02 Marks]

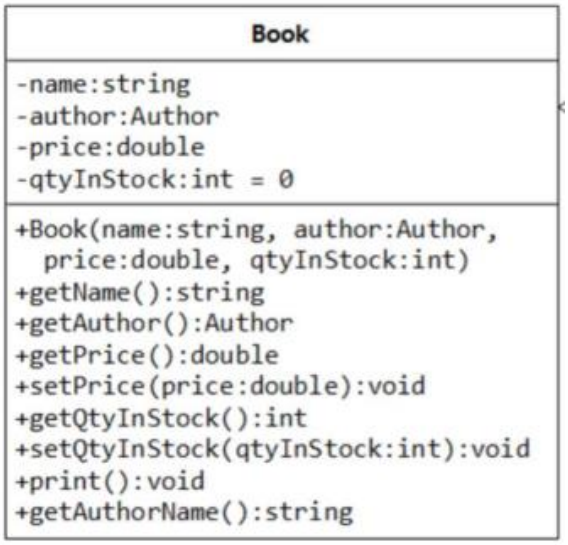
1)Abstraction. 2)Encapsulation. 3)Inheritance. 4)Polymorphism. [1]

Abstraction means displaying only essential information and hiding the details. Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation. **[1]**

Question # 2: [10 points] Suppose you have to create a class named as "Book". Question has four parts but read the following paragraph first.

You have to decide on what data members and member functions this "Book" class should have. Choose yourself what class members should be static or non-static and constant or non-constant. And also, you have to decide any friendship declaration if required. No driver program (main function) required in your answer.

i. Create a UML Class Diagram for "Book" class. [02 Marks]



ii. Write the Class Definition only. Class definition must match the UML Class Diagram. Only define the functions asked in part iii & iv. Definition of rest of functions not required. [02 Marks]

iii. Define a copy constructor [03 Marks]

iv. Overload the Stream Insertion Operator i.e **" >> "** [03 Marks]

```
#include<iostream>
using namespace std;
class Book
{
private:
   int ISBN;
   char* name;
   int price;
   int getSize(char* n)
   {
      int count=0;
      while(n[count]!='\0')
      {
         count++;
      }
      return count+1;
   }
public:
   Book()
   {
      ISBN=0;
      setName("");
      price=0;
   }
   Book(int i,char*n,int p)
   {
```

```
      ISBN=i;
      setName("");
      price=p;
   }
   Book(Book&b)
   {
      ISBN=b.ISBN;
      setName(b.name);
      price=b.price;
   }
   void setName(char* n)
   {
      int size=getSize(n);
      name=new char[size];

      for(int i=0; i<size; i++)
      {
         name[i]=n[i];
      }
   }
   void display()
   {
      cout<<ISBN<<endl;
      cout<<name<<endl;
      cout<<price<<endl;
   }
   friend istream& operator>>(istream& in,Book&b);
};
istream& operator>>(istream& in,Book&b)
{
   cin>>b.ISBN;
   cin>>b.name;
   cin>>b.price;
}
main()
{
   Book b;
   cin>>b;
   b.display();
   Book b1=b;
   b1.display();
}
```

Question # 3: [10points] Given **Watch.h** and **Source.cpp** (Main function) file of code, complete the code by writing **"Watch.cpp"**. Perform this task using deep copy if required.

Watch.h

```
Class Watch
{
        char *name;
        bool isDigital;
Public:                                  // points
        Watch ( );                        //1
        Watch (const char *, bool);       //2
        Watch (const Watch &);            //2
        ~Watch ( );                       //1
        void setName (const char *);      //2
        char* getName ( );                //2
}
```

Source.cpp

```
#include<iostream>
#include "Watch.h"
using namespace std;
int main()
{
   Watch    w1;
   Watch w2("Rolax", true);
   Watch  w3 = w2;
   cout << w3.getName();
  return 0;
}
```

```
class Watch
{
  char *name;
  bool isDigital;
  int getSize(const char* n)
  {
    int count=0;
    while(n[count]!='\0')
    {
      count++;
    }
    return count+1;
  }
  void setName(char* n)
  {
    int size=getSize(n);
    name=new char[size];

    for(int i=0; i<size; i++)
    {
      name[i]=n[i];
    }
  }
public:
  Watch ( )
  {
    setName("");
    isDigital=true;
  }
  Watch (const char *n, bool di)
  {
    setName(n);
    isDigital=di;
  }
  Watch (const Watch &other)
  {
    setName(other.name);
    isDigital=other.isDigital;
  }
  ~Watch ( )
  {
    if(!name)
      delete name;
  }
  void setName (const char *n)
  {
    int size=getSize(n);
    name=new char[size];

    for(int i=0; i<size; i++)
    {
      name[i]=n[i];
    }
  }
  char* getName ( ){
    return name;
}
};
```

Question # 4: [10 Points] Implement Distance.cpp (Function definition file) for the given UML class diagram of the Distance class. You must ensure that all member functions are properly defined in the Distance.cpp file in order to successfully implement the operators overloading as represented in the UML class diagram below.

Distance
- feet: int - inches: int
+ Distance(int=0, int =0) + showDistance( ): int + operator == (const Distance&): bool + operator + : Distance

```
class Distance{
   int feet;
   int inches;
public:
   Distance(int f=0,int i=0)
   {
      if(i>=12)
      {
         feet=i/12;
         inches=i%12;
         feet+=f;
      }
      else
      {
         feet=f;
         inches=i;
      }
   }
   void showDistance()
   {
      cout<<"Feet: "<<feet<<endl;
      cout<<"Inches: "<<inches<<endl;
   }
   bool operator==(const Distance& other)
   {
      if(feet==other.feet && inches==other.inches)
         return true;
      return false;
   }
   Distance operator+(Distance& other)
   {
      return Distance(feet+other.feet,inches+other.inches);
   }
};
int main()
{
   Distance d1(12,6);
   Distance d2(12,5);
   Distance d3=d1+d2;
   d3.showDistance();
   return 0;
}
```

Question # 5: [5 Points] Write output of the following code. Keep in mind there is no logical or syntax error in the code.

```
#include<iostream>
using namespace std;
class Donor {
private:
	int id;//exclusive
	static int pfund;//shared (allocated once)
public:
	Donor() { }
	Donor(int id) { this->id = id; }
	static int getpfund() { return pfund; }
	static void  contribute(int amt) { pfund = pfund + amt; }
};
int Donor::pfund = 1000;
int main() {
	cout << Donor::getpfund();
	Donor::contribute(10000);//ucp contributes 10000
	Donor owais(1), aqeel(2), umer(3), osama(4);
	owais.contribute(100);
	aqeel.contribute(100);
	umer.contribute(100);
	osama.contribute(0);
	cout << "owais wants to check Pfund =" << owais.getpfund() << endl;
	cout << "aqeel wants to check Pfund =" << aqeel.getpfund() << endl;
	umer.contribute(-200);
	cout << "osama wants to check Pfund =" << osama.getpfund() << endl;
	cout << "ucp wants to check Pfund =" << Donor::getpfund() << endl;
}
```