|  |
| --- |
| Serial No: |
| **1st Mid Term Exam** |
| **Total Time:1 Hour** |
| **Total Marks: 50** |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Signature of Invigilator |

|  |
| --- |
| **CS103 Computer Programming** |
| Saturday April 07, 2018 |
| **Course Instructor** |
| Maria Maqsood Asma Sattar Muhammad Atif |

|  |
| --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ . .  Roll No Section Signature |

## 17f-8195 B Muhammad usama.

## DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

**Instructions:**

1. Verify at the start of the exam that you have a total of Three (3) questions printed on eleven (11) pages including this title page.
2. Attempt all questions on the question-book and in the given order.
3. The exam is closed books, closed notes. Please see that the area in your threshold is free of any material classified as ‘useful in the paper’ or else there may a charge of cheating.
4. Read the questions carefully for clarity of context and understanding of meaning and make assumptions wherever required, for neither the invigilator will address your queries, nor the teacher/examiner will come to the examination hall for any assistance.
5. Fit in all your answers in the provided space. You may use extra space on the last page if required. If you do so, clearly mark question/part number on that page to avoid confusion.
6. Use only your own stationery and calculator. If you do not have your own calculator, use manual calculations.
7. Use only permanent ink-pens. Only the questions attempted with permanent ink-pens will be considered. Any part of paper done in lead pencil cannot be claimed for checking/rechecking.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Q-1 | Q-2 | Q-3 | Total |
| **Total**  **Marks** | 30 | 10 | 10 | **50** |
| **Marks Obtained** |  |  |  |  |

**Vetted By: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Vetter Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **University Answer Sheet Required:** | **No** |  |  |  |  | **Yes** |  |

|  |  |
| --- | --- |
| **Question 1** | **Marks 30** |

**Write the output of the following codes. If there is an error clearly highlight the error and after correcting the error write output.**

class A

{public:

A(){ a++;}

static int a;

void show(){cout<<" I am A "<<a <<endl;}

};

int A:: a=3;

class B : public A{

public:

static int b;

B(){a--; b++;}

void show() {cout<<" I am B " <<b<<endl; A::show(); }

};

int B:: b=2;

int main()

{

B b1[3]; B b2; B \*bptr=new B;

bptr->show();

A a1[3]; A a2; A \*aptr=new A;

aptr->show();

return 0;

}

|  |
| --- |
| **I am B 7**  **I am A 3**  **I am A 8** |

class A

{

public:

int\* ptr;

A() { ptr=new int(3); }

~A(){ delete ptr; ptr=NULL;}

};

void fun(A obj)

{cout<< \*(obj.ptr);}

int main()

{ A a;

fun(a);

fun(a);

return 0;}

|  |
| --- |
| **3(garbage value occurred)** |

class A

{ int a;

public:

A(){cout<<"A's Default"<<endl;}

A(int var):a(var)

{cout<<"A's constructor' "<<endl; }

void show(){cout<<"A"<<endl;}

~A() {cout<<"A"<<endl;}

};

class B : private A

{int b;

public:

B(int v)

{b=0;

cout<<"B's Constructor"<<endl;}

void show(){cout<<"B"<<endl;}

~B() {cout<<"B"<<endl;}

};

class C

{public:

C() { cout<<"C's Default"<<endl;}

void display(){cout<<"C"<<endl;}

~C() {cout<<"C"<<endl;}

};

class D : public C,public B

{public:

D():B(2) {cout<<"D's Constructor"<<endl;}

void fun(){cout<<"D"<<endl;}

~D() {cout<<"D"<<endl;}

};

int main()

{ D d1;

d1.show();

d1.B::show();

return 0;

}

|  |
| --- |
| **C’s Default**  **A’s Default**  **B’s Constructor**  **D’s Constructor**  **B**  **B**  **B Destructor** |

class A

{public:

void show(){cout<<" I am A";}

};

class B:public A

{

};

class C:public B, private A

{

};

int main()

{

C c1; c1.show();

return 0;

}

|  |
| --- |
| **I am A** |

class A {

private :

int data;

public:

A(){data=5;}

void fun()

{ cout<<"Hello world";

}

void print()const

{data=8;

cout<<data;

}

};

int main() {

const A a1;

a1.fun();

return 0;

}

|  |
| --- |
| **Hello world** |

class A

{

    int x;

public:

    A() { cout << "A's constructor called " << endl;  }

};

class B

{

    static A a;

public:

    B() { cout << "B's constructor called " << endl; }

    static A getA() { return a; }

};

A B::a;  // definition of a

int main()

{

    B b1, b2, b3;

    A a = b1.getA();

    return 0;}

|  |
| --- |
| **A’s constructor called**  **B’s constructor called**  **B’s constructor called**  **B’s constructor called** |

class Box

{

double width;

public:

friend void printWidth( Box box );

void setWidth( double wid );

};

void Box::setWidth( double wid )

{

width = wid;

}

void printWidth( Box box )

{

box.width = box.width \* 2;

cout << "Width of box : " << box.width << endl;

}

int main( )

{

Box box;

box.setWidth(10.0);

printWidth( box );

return 0;

}

|  |
| --- |
| **Width of box :20** |

class square;

class rectangle

{

int width, height;

public:

int area ()

{

return (width \* height);

}

void convert (square a);

friend class square;

};

class square

{

private:

int side;

public:

void set\_side (int a)

{

side = a;

}

};

void rectangle::convert (square a)

{

width = a.side;

height = a.side;

}

int main ()

{

square sqr;

rectangle rect;

sqr.set\_side(6);

rect.convert(sqr);

cout << rect.area();

return 0;

}

|  |
| --- |
| **36** |

class A

{

static int x;

int y;

public:

static void SetData(int xx)

{

x = xx;

}

static void Display()

{

cout<< x<<y;

}

};

int A::x = 0;

int main()

{ A obj;

obj.SetData(33);

A::Display();

return 0;

}

|  |
| --- |
| **33** |

class A

{

int x;

public:

A() { this->x=0; }

void increment()

{this->x++; }

void print()

{cout<<this->x<<endl;

}

A fun()

{x=5;

return \*this;

}

};

int main()

{ A obj;

obj.increment();

obj.print();

obj.fun().increment();

obj.print();

return 0;}

|  |
| --- |
| **1**  **5** |

|  |  |
| --- | --- |
| **Question 2** | **Marks 10** |

1. What are the three situations in which the copy constructor is called?

|  |
| --- |
| **In these situation copy constructor called.**  **Object1=object2;**  **Object1(2);**  **Return object1;** |

1. Differentiate between function overloading and overriding.

|  |
| --- |
| **Overloading :**  **Function overloading when two function have same name and different parameter list.**  **Overriding :**  **Function of base class is override when derive class function have (1) same name (2) same parameter list (3)same declaration type.** |

1. What is member initialization list? When we use member initialization lists in constructors?

|  |
| --- |
| **Function non static member function. It is use to initialize non static object initialization list of constructor include non-static variable initialization. And it also use to initialize sub objects.** |

1. Define constant data member and constant member function with the help of example?

|  |
| --- |
| 1. #include<iostream>   using namespace std;   1. class my 2. { 3. const int c; //we cannot change the value of c 4. int x; 5. my(): c(0) 6. { } 7. public: 8. void co(const my m,my n) const 9. { 10. //x=0; //illegal 11. //m.x=0; //illegal 12. n.x=0; //legal 13. } 14. }; 15. int main() 16. { 17. system("pause"); 18. return 0; 19. } |

1. What do you mean by public, private and protected inheritance?

|  |
| --- |
| **Public inheritance:**  **In this inheritances all public and protected data member of base class behave as public data member of derive class.**  **Private:**  **In this inheritances all public and protected data member of base class behave as private data member of derive class.**  **Protected inheritance:**  **In this inheritances all public and protected data member of base class behave as protected data member of derive class.** |

|  |  |
| --- | --- |
| **Question 3** | **Marks 10** |

Design various classes and write a program to computerize the billing system of a hospital.

1. Design the **class PersonType** that has following members : First name, Last name, getname(), setname()
2. Design the **class doctorType,** inherited from the class personType, with an additional data member to store a doctor’s speciality. Add appropriate constructors and member functions to initialize, access, and manipulate the data members.
3. Design the **class patientType**, inherited from the class personType, with additional data members to store a patient’s ID, age, date of birth, attending physician’s name, the date when the patient was admitted in the hospital, and the date when the patient was discharged from the hospital. (Use the class doctorType to store the attending physician’s name.)
4. Design the **class billType** with data members to store a a patient’s hospital charges, such as pharmacy charges for medicine, doctor’s fee, and room charges (Charge per day is 1000 Rs. (Patient Discharge date - Patient admitted date )\* 1000 = total room charges). Add appropriate constructors and member functions to initialize and access and manipulate the data members. Add appropriate inheritance relationship between **class billType and class patientType.**

Add appropriate constructors and member functions to initialize, access, and manipulate the data members. Write a program to test your classes.

|  |
| --- |
|  |

---------------------------------------------END-----------------------------------------------

**Extra**

NOT INC:UDED:

REPEAT (5)

#include<iostream>

using namespace std;

class Test {

int value;

public:

Test(int v = 0) {value = v;}

int getValue() {return value;}

};

int main() {

const Test t;

cout << t.getValue();

return 0;

}

What does compiler do when it sees a keyword friend before any function definition?

(already 3 questions related to friend)

Ans :Friend function define out of scope of class. Friend functions are not in class scope.

Can we inherit the constructor and destructor of base class

(covered)

Ans : we cannot inherit constructor and destructor of base class. Constructor of base class is called before when constructor of derive class is called.

How to overload a class destructor? Give proper explanation.???

Ans :: We cannot overload the destructor. Because destructor is only use to free the dynamic memory. So, we don’t need to overload the destructor.