B.Sc. Engg. / HD CSE 3rd Semester (67) 15 March 2016 (Morning)

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)
ORGANISATION OF ISLAMIC COOPERATION (OIC)
Department of Computer Science and Engineering (CSE)

MID SEMESTER EXAMINATION **WINTER SEMESTER, 2015-2016**

DURATION: 1 Hour 30 Minutes **FULL MARKS: 75**

CSE 4301: Object Oriented Programming

Programmable calculators are not allowed. Do not write anything on the question paper.

There are 4 (four) questions. Answer any 3 (three) of them.

Figures in the right margin indicate marks.

---

1. a) Create a class called `MyString` which has an array of integers called `intAr`. Initialize the class with three different constructors. The default constructor allocates memory of size `10` for `intAr`. Another Constructor takes the size of the array `intAr` as parameter and dynamically allocates memory for `intAr`. The final constructor takes another array of integers along with its size as parameters and initializes `intAr` with it. The destructor of `MyString` frees the memory allocated for `intAr`. The class has the following methods: 18

One method called `SetElement` takes an integer as input parameter and adds it to the array `IntAr`. This method will print a message to indicate memory of the array `IntAr` is full.

Another method called `EditElement` will take an index and an element as input parameters and set the element at the desired index of the array `IntAr`. A message will be shown to the users if the index is out of bounds.

Another method called `ConcatStr` will accept an object pointer of class `MyString` and dynamically concatenate its contents after the elements of `IntAr`.

Another method `ReverStr` reverses the elements in the array `IntAr`.

The final method called `PrintStr` will print the ASCII equivalent characters of these integers.

Notes:

`IntAr` can only accept values ranging from 1 to 255. Otherwise, the value will be discarded and error message will be generated.

You can include other variables to the class `MyString` to execute the required tasks.

b) With proper example discuss different types of ambiguity in Function Overloading. 4

c) Will the following code segment generate any error? Why? 3

```
void foo(int a , int b=0, int c=0 ){

}
void foo(int a){

}
int foo(int PassedInteger){
      return PassedInteger;
}
```

Figure 1: Code for question 1.(c)

2. a) Will the following code generate any exception? If any, why? Also discuss, if there are errors, how can we correct some, if not all of them? 6

```
class SampleClass{
      char *p;
public:
      SampleClass(char *passedP){
            p=new char[strlen(passedP)];
            strcpy(p,passedP);
      }
      ~SampleClass(){
            cout << "Destroying P: " << p << endl;
            free(p);
      }
     void show(){cout<<p<<endl;}
};
void foo(SampleClass samp){samp.show();}
int main(){
      SampleClass sc("Hello");
      sc.show();
      if (true){
            SampleClass sc2("World");
            sc2.show();
            sc = sc2;
      }
      sc.show();
      foo(sc);
      return 0;
}
```

Figure 2: Code for question 2.(a)

b) Write a program that includes a class named **Currency**. The class has a **private** variable named **amount** of type **double**. You can set the value of the variable **amount** with the function named **Set_Amount**. But there can be no function in the class that returns the value of **amount**. The class cannot have any **public** variables either. 16

Your program should include another class called **Calculator** which has a **public** overloaded method called **Get_Amount**. This class also have two **private** variable called **round** of type integer and **fraction** of type float. **Get_Amount** can accept one or two parameters of type **Currency**. The method **Get_Amount** should have the capability to access the variable **amount** of **Currency** class.

The task of the **Get_Amount** method that accepts only one parameter is to save the **floor** value of the variable amount into the variable **round**. Also the fractional part of the variable amount should be stored in the variable **fraction**. For example if the value of variable amount is 12.345, the value of round should be 12 and the value of fraction should be 0.345.

The method **Get_Amount** which accepts two parameters takes the **floor** values from both the object of class Currency, adds them together and save them in the variable **round**. Also the fractional values are added and saved in variable **fraction**.

The class **Calculator** cannot have any other methods or variables. You should write a function outside of these two classes that can access the private variables in **Calculator** class and print their values.

c) What would be the output of the following program? 3

```
int x=5;
int &foo(){
      x = 30;
      return x;
}
int main(){
      foo() = foo() + 3;
      cout << x << endl;
      return 0;
}
```

Figure 3: Code for question 2.(c)

3. a) Create a program to handle Point of Sales of a restaurant. An admin of the restaurant adds new categories of food items in the menu. The information that are required to add a category are Name and Description of the category. An Admin can also Add Food Items under specific Category. Each Food Item should include Name, Description and Price. Moreover, an Admin can also add new Waiters and Cashiers in the system. Waiters can see the list of Categories and Food Items and place Orders. Orders include a list of Food Items and Total Price and Date of Placing Order. A Cashier can see how much money the restaurant earned on a specific date. A Date includes a Day, a Month and a Year entry. A User of your program has to at first select what type of user s/he is and according to the choice new options will appear in the screen. Create appropriate classes according to the requirements. To refer other classes, use the objects of the classes. In order to demonstrate your program, create different types of users, create food categories and food items. Also place food orders and retrieve sales record on a specific date. 15

b) *Find out the errors in the following program. Comment on why the errors are occurring. Re-write the program and find out the output of it.* 10

```
class ErrorCheck
{
      int *a;
public:
      ErrorCheck(int *a){a = a;}
      ~ErrorCheck(){cout << *a << endl;}
      int get_val(int idx){return a[idx];}
      void set_arr(int *a){a = a;}
};
int main(){
      int ar[] = { 1, 2, 3 };
      ErrorCheck erArr[3]= { ar, ar, ar };
      int i=0;
      for (ErrorCheck ep = erArr; i<3; i++){
            cout<<ep.get_val(i)<<endl;
            ep++;
      }
      ErrorCheck erArr2 = new ErrorCheck;
      ErrorCheck erArr3 = new ErrorCheck[ar];
      for (i = 0; i < 3; i++)
            cout << erArr3->get_val(i) << endl;
      return 0;
}
```

Figure 4: Code for question 3.(b)

4. a) Create a class Called **MyArray** which can dynamically allocate memory for an integer array or a character array with a specific size provided by the user. The class should have overloaded methods to add new elements in specific type of array and retrieve them when user provides appropriate index. Your program should generate message if users want to insert or retrieve elements from out-of-bound index of the arrays. Also the users should get error message when they want to add elements in an invalid index of the array. The destructor of the class should free all the memory allocated for the array of the class. 16

There should be another class called **ArrayManipulation** where you should include methods that take inputs from the users to put values of specific type of Array or retrieve them in the **MyArray** class. You cannot declare any instance of class **MyArray** inside this class. Inside main function, you should declare two instances of **MyArray** class and take the size and type of array as input from the user. Then you should declare an instance of **ArrayManipulation** class and use the methods inside the class to add or retrieve elements from the passed instance of **MyArray** class.

Note: you cannot take any user input inside main function and you cannot have any other functions outside the aforementioned two classes.

b) What would be the output of the following program? 6

```
class SampleClass
{
int i, j;
public:
            SampleClass():i(0), j(0){}
            SampleClass(int a, int b):i(a), j(a+b){}
            ~SampleClass()
            {
                  cout << i << ' ' << j<<endl;
            }
};
int main(){
      SampleClass sc1(1,1),sc2(2,2),sc3(3,3);
      SampleClass scArr[3] = {SampleClass(10,10),
      SampleClass(),SampleClass(30, 30)};
      return 0;
}
```

Figure 5: Code for question 4.(b)

c) What are the three defining traits of Object Oriented Programming? 3