

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

**MID SEMESTER EXAMINATION**  
**DURATION: 1 HOUR 30 MINUTES**

**WINTER SEMESTER, 2022-2023**  
**FULL MARKS: 75**

**CSE 4301: Object Oriented Programming**

**Programmable calculators are not allowed. Do not write anything on the question paper.**  
 Answer all 3 (three) questions. Figures in the right margin indicate full marks of questions whereas corresponding CO and PO are written within parentheses.

1. a) Why the declaration presented in the first line is better than the second one in the Code Snippet 1. 3  
(CO1)  
(PO1)
- ```
1 enum isWord{ No, Yes};
2 enum isWord{ Yes, No};
```
- Code Snippet 1:** Two different declaration of enum type for Question 1. a)
- b) If a function is very brief, the instructions required to call it can take up as much space as the instructions contained within the function body, incurring a space penalty in addition to a time penalty. What kind of approach would solve this issue without repeating the same code? Explain the approach with an example. 3  
(CO1)  
(PO1)
- c) Normally while passing an object as an argument to a function, a reference argument is used. 2 + 1  
(CO1)  
(PO1)
- i. Explain the reason for doing this.
  - ii. How can the programmer ensure that the value passed as a reference argument will not be changed by the calling function?
- d) Sometimes, the programmers want to call functions without providing all their arguments. Explain the process with an example code. What are the restrictions you should remember to use this feature? 3  
(CO1)  
(PO1)
- e) Discuss the correctness of the Code Snippet 2. 3  
(CO1)  
(PO1)
- ```
1 ints func(int p, int q){
2     int a = p + q;
3     return a;
4 }
```
- Code Snippet 2:** A function for Question 1. e)
- f) Show the difference between assignment operation and initialization operation with an example code. What are the prototypes of the member function for these two operations? 3  
(CO1)  
(PO1)
- g) Differentiate between are the default constructor and the default copy constructor functions. 3  
(CO1)  
(PO1)
- h) Discuss the use of initializer list. 3  
(CO1)  
(PO1)
- i) Define is a friend function. Should a program contain a lot of friend functions? Justify. 3  
(CO1)  
(PO1)

2. a) Write a class **Date** which stores the day, month, and year. The default date is 01 January, 1970. One cannot assign an invalid date (i.e.  $1 \leq \text{day} \leq 31$ ,  $1 \leq \text{month} \leq 12$ , and  $\text{year} > 0$ ). If anyone tries to set any invalid date, it will keep the previously stored date. Add any necessary function to run the following code. 15  
(CO2)  
(PO2)

```

1 int main() {
2     Date today(2023, 10, 04), tomorrow(2023, 10, 05), dob(1990, 03,
3         26), error(1990, 03, 32);
4     if(today > dob)
5         cout << today << " is greater than " << tomorrow << endl;
6     if(today < tomorrow)
7         cout << today << " is less than " << tomorrow << endl;
8     if(today == today)
9         cout << today << " is same as " << today << endl;
10    cout << error << endl;
11    return 0; }

```

**Code Snippet 3:** Driver function for Date class for Question 2. a)

Note that the output of the Code Snippet 3 is in Table 1

4-10-2023 is greater than 5-10-2023
4-10-2023 is less than 5-10-2023
4-10-2023 is same as 4-10-2023
1-1-1970

**Table 1:** Output of Code Snippet 3 for Question 2. a)

- b) Update the solution for Question 2. a) in such a way that it can check the validity of the value of the day based on the month. February has 28 days and 29 days if it is a leap year. January, March, May, July, August, October, and December have 31 days. The rest of the month has 30 days. A leap year occurs every four years (those whose number is divisible by four) except for century years whose number is not divisible by 400. 7  
(CO2)  
(PO2)
- c) Which of the member functions of the **Date** class mentioned in Question 2. a) needs to be converted to a constant member function? 3  
(CO2)  
(PO2)
3. a) A **Person** class represents a human being. Data members should include the person's name, year of birth, relation, academic qualification, job status. Include a constructor, a destructor, access functions, and a print function. For the national census program, this class will be used. A census officer will go to each house and take information about all the members of the house. First, the officer gives the input of the number of members in the house and then fills up the values in an object of person for each house member. Later all the information will be displayed and write a program that will be used by the census officer. Implement the **Person** class. 9  
(CO2)  
(PO2)
- b) Write a class named **ComplexNumber** which stores a complex number consisting of real part and imaginary part. We express a complex number as  $a + ib$  where  $a$  is real and  $b$  is the imaginary part. Add a member function that returns the modulus of a complex number. Overload the operators i.e. relational operator ( $>$ ,  $<$ ,  $==$ ,  $>=$ ,  $<=$ ,  $!=$ ) and arithmetic operators ( $+$  and  $-$ ) so that these binary operator can be applied on two objects of **ComplexNumber** class in conventional way. 7  
(CO2)  
(PO2)

- c) Write a **MyString** class that mimics the standard c++ string class. It uses a char array member variable to store. Add necessary member functions to the class that allows the code mentioned in Code Snippet 4 run.

7  
(CO2)  
(PO2)

```
1 int main(){
2     MyString s1;
3     char xstr[] = "Faisal";
4     s1 = xstr;
5     cout<<s1; // this prints Faisal
6     s1 = "Mr. " + s1;
7     s1 += " hussain";
8     cout<<s1; // this prints Mr. Faisal hussain
9     s1[10] = 'H';
10    cout<<s1; // this prints Mr. Faisal Hussain
```

**Code Snippet 4:** Testing MyString Class. Question 3. c)