RollNo:

Thapar Institute of Engineering & Technology, Patiala
Computer Science & Engineering Department
Mid Semester Examination (MST)

B. E. (First Year): Sem-II (2022-23) **Course Code:** UTA018 (Object Oriented Programming)

Date: 7-Mar 2023 **Time:** 2 Hours **M.Marks:** 25

Faculty: Raman Goyal, Jasvinder Pal Singh, Ravneet Kaur, Saif Nalband, Neenu Garg, Seemu Sharma, Deep Mann, Aditi Sharma, Amrita Dahiya, Naveen Kumar

NOTE: Solve problems **IN ORDER,** otherwise they WILL **NOT** be **checked**. If you are not sure about the current question then leave appropriate pages and move on. **New page** for new problem. Do not use pencil for answering the questions, otherwise such answers will not be evaluated .

Q1. "Friend functions can access private and protected members of a class but so do the member functions of a class". Then, what is the advantage of a friend function over a member function? **(1 mark)**
Write a program in C++ to describe the advantage of friend function over member functions. **(4 marks)**

Q2. (a) List different types of inheritance with their syntax. Draw suitable diagrams for each type. **(4 marks)**
(b) Explain the working of copy constructor with suitable example **(1 mark).**

Q3. (a) In the given code perform following tasks. **(2 marks)**
- **i.** Fill the line 24 and
- **ii.** Fill the line 27 to initialize static variable with value 1 before object creation.

```
1.  #include<iostream>
2.  using namespace std;
3.  class Test{
4.  static int i;
5.  int num;
6.  public:
7.  void static getvalue(int);
8.  Test(int x)
9.  {
10.         num=x;
11.         }
12.         void show()
13.         {
14.         cout<<"i="<<i<<endl;
15.         cout<<"num="<<num<<endl;
16.         }
17.         };
18.         void  Test :: getvalue(int x)
19.         {
20.         i = x;
21.         i++;
22.         cout<<i<<endl;
23.         }
24.         _____;
25.         int main()
26.         {
27.         _____;
28.         Test T1(10);
29.         T1.show();
30.         }
```

Q3 (b) Write a program in C++ having the following properties: **(3 marks)**

- Define a class having two data variables of integer type in private scope.
- Array of objects should be declared in main function dynamically (using **new** operator)
- Input the data variables using a member function named as **void set_data()** in public scope.
- Increment the values of data variables by one using a member function named **void update_data()** in public scope having array of object as an argument.
- For displaying the updated value of data variables, create a member function named as **void output()** in public scope.

Q4. Write a program in C++ having a class *Tracker* that tracks the footsteps of a user. Tracker should have three private data members: *PersonId, step_count, stair_count*. The class should satisfy the following requirements: **(5 marks)**

(a) A parameterized **constructor** should be defined to initialize data members. *PersonId* should be input by the user, whereas *step_count* and *stair_count* if not known may be set as 0 using constructors with default arguments.

(b) A member function *incrementCount*(int **n**) to be **defined outside the class** that increment the *step_count* and *stair_count* by the value n.

i.e.
```
step_count = step_count + n;
stair_count = stair_count + n;
```

(c) A **non-member function** *convertStep()* should be provided that converts *stair_count* to *step_count* (Assuming 1 stair_count = 2 step_count).

(d) A *displayCount()* function should be provided that displays the *PersonId* and updated *step_count* (after increment and conversion of stair_count to step_count).

(e) In addition, the class must use **array of objects** for creating the objects and track how many Tracker objects are created. A means for querying this count should be provided. The code must not use any global variables. (Hint: Use **static members** and **static functions** to display the count of Tracker objects.)

Q5. Write short notes on the following:

(a) Namespace **(2 marks)**
(b) Encapsulation **(1 mark)**
(c) Advantages of static member functions **(2 marks)**