C++ Programming (2) - CSC 1201
2nd term Year 2019 /2020
Worksheet (7)

King Saud University
Natural Sciences And
Engineering Program
Programming And Database
Diploma

Student Name:	Student Number: .....

Q1: Tracing the output.

```
      //Assume the address of nValue is equal to 2333
int main(){
double nValue = 7;
double *pnPtr = &nValue;

cout << pnPtr << endl;
cout << pnPtr+1 << endl;
cout << pnPtr+2 << endl;
cout << pnPtr+3 << endl;
}
```

OUTPUT:

```
0x7ffeefbff530
0x7ffeefbff538
0x7ffeefbff540
0x7ffeefbff548
```

Q2: Find the errors

```
int nValue = 5;
double dValue = 7.0;

int *nPtr = &nValue;
double *dPtr = &dValue;
nPtr = &dValue;
dPtr = &nValue;
```

Solve:

```
nPtr = &dValue;
dPtr = &nValue;
```
beacues its different data type.

Q3:Trace the following program and write the output:

```
#include<iostream>
using namespace std;
////////////////////////////////
class Person {
   // Data members of person
public:
    Person(int x)  { cout << "Person::Person(int ) called" << endl;   }
};

class Faculty : public Person {
   // data members of Faculty
public:
    Faculty(int x):Person(x)   {
       cout<<"Faculty::Faculty(int ) called"<< endl;
    }
};

class Student : public Person {
   // data members of Student
public:
    Student(int x):Person(x) {
        cout<<"Student::Student(int ) called"<< endl;
    }
};

class TA : public Faculty, public Student  {
public:
    TA(int x):Student(x), Faculty(x)   {
        cout<<"TA::TA(int ) called"<< endl;
    }
};
 void main()  {
    TA ta1(30);
}
```

OUTPUT:

```
Person::Person(int ) called
Faculty::Faculty(int ) called
Person::Person(int ) called
Student::Student(int ) called
TA::TA(int ) called
```

```cpp
#include <iostream>
using namespace std;
int main() {
    int *pc, c;

    c = 5;
    cout << "Address of c (&c): " << &c << endl;
    cout << "Value of c (c): " << c << endl << endl;

    pc = &c;    // Pointer pc holds the memory address of
variable c
    cout << "Address that pointer pc holds (pc): "<< pc << endl;
    cout << "Content of the address pointer pc holds (*pc): " <<
*pc << endl << endl;

    c = 11;    // The content inside memory address &c is changed
from 5 to 11.
    cout << "Address pointer pc holds (pc): " << pc << endl;
    cout << "Content of the address pointer pc holds (*pc): " <<
*pc << endl << endl;

    *pc = 2;
    cout << "Address of c (&c): " << &c << endl;
    cout << "Value of c (c): " << c << endl << endl;

    return 0;
}
```

OUTPUT:

```
Address of c (&c): 0x7ffeefbff52c
Value of c (c): 5

Address that pointer pc holds (pc): 0x7ffeefbff52c
Content of the address pointer pc holds (*pc): 5

Address pointer pc holds (pc): 0x7ffeefbff52c
Content of the address pointer pc holds (*pc): 11

Address of c (&c): 0x7ffeefbff52c
Value of c (c): 2
```

```cpp
// C++ program to illustrate Pointer Arithmetic in C++
#include <iostream>
using namespace std;
void geeks()
{
    //Declare an array
    int v[3] = {10, 100, 200};

    //declare pointer variable
    int *ptr;

    //Assign the address of v[0] to ptr
    ptr = v;

    for (int i = 0; i < 3; i++)
    {
            cout << "Value at ptr = " << ptr << "\n";
            cout << "Value at *ptr = " << *ptr << "\n";

            // Increment pointer ptr by 1
            ptr++;
    }
}

//Driver program
int main()
{
    geeks();
}
```

OUTPUT:

```
Value at ptr = 0x7ffeefbff51c
Value at *ptr = 10
Value at ptr = 0x7ffeefbff520
Value at *ptr = 100
Value at ptr = 0x7ffeefbff524
Value at *ptr = 200
```