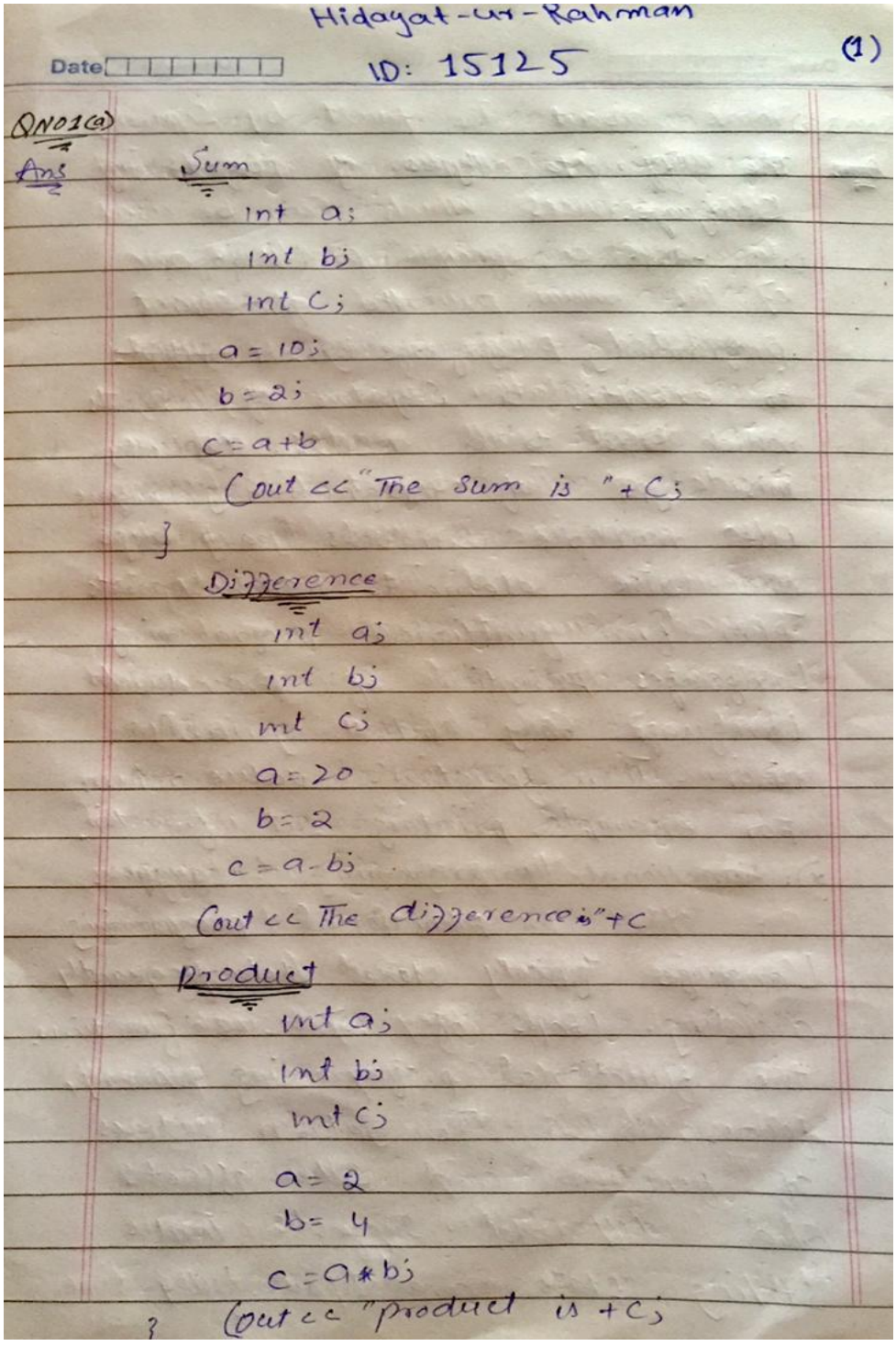
NAME:HIDAYAT UR RAHMAN
ID:15125
BS (SE)
SUB:FROGRAMING FUNDAMENTALS
SIR:DR.FAZAL-E-MALIK



Hidayat-ur-Rahman (1)
Date
ID: 15125

QNO1(a)
Ans

Sum

```
int a;
int b;
int c;
a = 10;
b = 2;
c = a+b
Cout << "The sum is " + c;
}
```

Difference

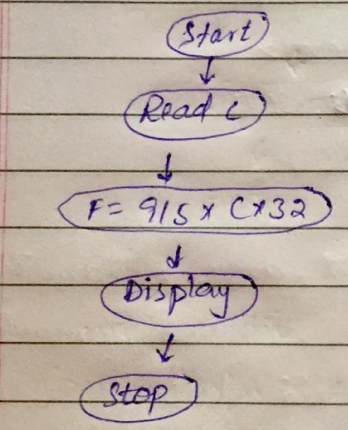
```
int a;
int b;
int c;
a = 20
b = 2
c = a-b;
Cout << The difference is" + c
```

Product

```
int a;
int b;
int c;
a = 2
b = 4
c = a*b;
} Cout << "product is + c;
```

Date: Day M T W T F S

Q No 1 (b)



X —— X —— X —— X

QNO2(a):

Ans

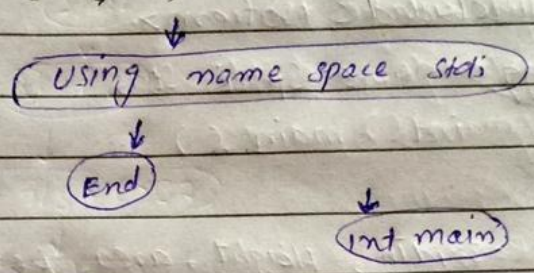
```
# included <isotrem>
using name Space Std;
    int main ()
{
 in width, Height, area, perameter,
 Cout << \n\n find area & perameter
   of rectengle :\n";
 Cout << " input The hight of rectand:";
 Cin >> Height";
 (out << " input The width of
rectengle
 Cin >> " width
  area = (Height * width);
  perameter = 2 * (Height + width);
  Cout << " The area of The rectengles
  is ;" << perimetus << endo;
   Cout << endo;
    return 0;
  {
```

Out put:

→ Height of Rectengle = 10
→ width of Rectengle = 15
→ The area of Rectengle = 50

Date Day M T W T F S

Flow Chart :-




```
int width Height, area, peri;
Cout << "\n\n Find area & peri of
rectangle (c:)s,";
Cout << " " " " \n";
Cin) >> Height
Cout << " input the width of
 rectangle : ";
Cin) >> width
area = ( Height + width)
Peri = 2 ( Height + width);
Cout << " the area of Rectangle
is " << area << end;
Cout << " the peri of rectangle : " <<
Peri << end;
Cout << end;
   return 0;
```

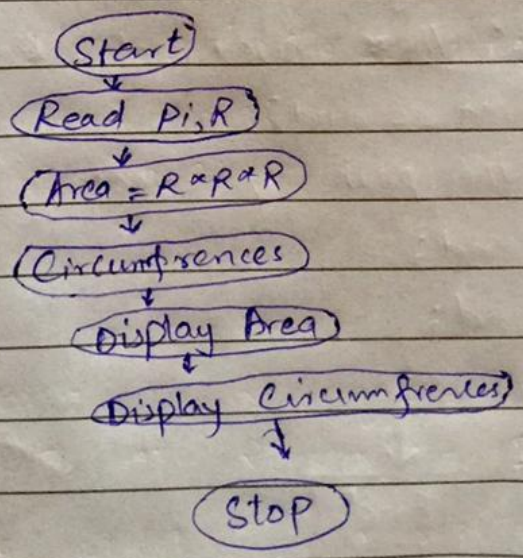
End

Date ☐☐☐☐☐☐☐☐ Day M T W T F S

Q NO 2 (b):-

Ans

```
1. #include <iostream>
2. Using namespace std;
3. int main ()
4. {
5. Int R=5;
6. float pi = 3.14;
7. float area;
8. float Circuference;
9. Area = pi * R * R;
10. Cout << area of Cicale is = ";
11. Cout << area;
12. Cout << \ncaircumference is = ";
13. Circumference = 2*pi*R;
14. Cout << ciscumfrence;
15 }
```



Date Day M T W T F S

Q.No:3 (a)

Ans: Programming Languages:

* "Programming language specially developed so that you could pass your data and instruction to the computer to do specific job.
* there are two major types of programming Languages
  - → Low Level Languages.
  - → High Level Languages.
* Low Level languages are further divided in to machine language and Assembly Language.
* High level languages. are for scientific application FORTRAN and c languages are used. on the other hand COBOL is used for business application.

→ Machine Language:

* "machine Language is the only Language that is directly understood by the computer. it dose not used any translator program.
* the only advantage is the program of machine Language. run very fast.

Date

* there is nothing "below" machine Language - only hardware.
* Impossible for humman to red. Consists of only 0's and 1's
  * 0001001111110000
* In the earliest days of computers, the only programming Languagees as avaliable were machine Languages. Each Computer had its own machine Language which was made of Streams of 0s and 1s.

⇒ Assembly Language::

* the next evolution programming Come with the idea of replacing binary code for instruction and addiesses with Symbols. Becuse thy used Symbols. these Language were first known as Symbolic Languages. the set of these mnemonic Languages were later refered to as assembly Languages.
* it is the first to improve the programming structure. you should know that Computer can handle

Date Day M T W T F S

numbers and letter.

* the set of symbols and letters form the assembly language and a translator program is required to translate the Assembly language to machine language
* this translator program used for assembly language is called Assembler.

⇒ Assembly Language:

* To program in assembly you need to understand concepts behind machine language and execution-fetch cycle of CPU.
* Assembly is a machine specific language.
* Although Assembly and machine language might look similar, they are in fact two different types of language.
* Assembly consist of both binary and simple words.
* machine code composed only of 0's and 1's

⇒ High Level Language:

" Although assembly language greatly improved programming

Date Day M T W T F S

efficiency. they still required programmer to concentrate on the hardware they were using. working with symbolic languages was also very tedious, because each machine instruction had to be individual coded. the desire to improve programmer efficiency and to change the focus from the computer to the problem being solved led to the development of high-level languages.

* Assembly and machine level language requires deep knowledge of computer hardware whereas in higher language you have to know only the instruction in english word and logic of the problem.
* Higher level languages are simple language that use english and mathematical symbols like. +, -, % etc for its program construction.
* Any higher level language has to converted to machine language for the computer to understand

Date: Day M T W T F S

* For example COBol (Common Business oriented language), FORTRAN (Formula Translation) and BASIC (Beginners All-purpose Symbolic instruction code) and high Level Languages.

Advantages of High Level Languages.

* Higher Level languages have a major advantage over machine and assembly language that higher Level language are easy to learn and use (Similar to the languages used by us in our day to day life.

x — x — x — x

The different types of programming Language.

I): procedural programming Langüge:

The procedural programming Language is used ~~multiple~~ to executes a Sequance of statments which lead to a result. Typically, This type of programming language use multiple variabels, heavy loops and othe elements which separats them from functional programming Languages. Function of procedural language may controll variables other then function's value return For example, printing out information.

II): Functional programming Language:

Functional programming Language typically store data, frequantly avoiding loops of recursive functions. The functional programming primry focous is on the return values of function and side effects and different suggests that storing state are powerfully

discouraged. For example, in an exceeding pure useful Language if a function is termed it expected that the function not modify or perform any o/p. it may, however, build algorithmic Calls. Functional languages are usually easier and build at easire to figure on obstract issue. however, they'll even be further from the machine "Thorem their Programming model makes it difficult to know precisely but the code is decoded into machine language (which are often problematic for System programming

III). Object-oriented Programming Language:-

The programming Language views the world as a group of objects that have internal data and external ~~data~~ assessing parts of the data. The aim The programming Languages is to

Date: Day M T W T F S

Think about the fault by Separating it into a collection of objects that offer services which can be used to solve a specific problume one of the main principal of object orientated programming languges is encapsulation that everything an object will need must be inside of the obeject. This language also emphisize reusability through inheritance and the capacity to Spread Current implementation without having to change a great deal of code by using polymo-rphism.

IV): Scripitng Programming Languages:
the programming Languages are often procedural and may comprise object-oriented languages elements, but they fall into their own category as they are normally not full-programming languages with support for devolpment of large system. For example, they may have not compile-time

type checking. Usually, These language require tiny to get Started.

v): Logic programming Language:-

These types of language let programmers make declarative Statements and Then allow The machine to reason about the consequanses of those Statements and ~~Then allow the machine and~~ asense This language doesn't tell the Computer how to do Some Thing, but employing restrictions on what it must consider doing.

X —— X —— X —— X

Date ☐☐☐☐☐☐☐☐ Day M T W T F S

Q NO:3 (b)

Ans:- * Complir is a program translator that translates the instruction of a high level Language to machine language

* it is called Complir becuse it complies machine Language instruction for every program Instruction of high level Language.

* thus Compler is a program translator Like assembler but more Sophisticated. it scane the entire program first and then translates it in to machine code.

* the programs written by the programmers in high level language is called source program. After this program is converted to machine Language by the Complirs it is called object program.

* A Complirs can translates only those source programs which heave been written, is that Language.

X ——— X ——— X

Date

Interpreter:-

* An interpreter is another type of program translator used for translating higher level language in to machine language.
* It takes one statement of higher level language, translate it into machine language and immediatly execute it.
* Translation and execution are carried out for each statement.
* it differs from compiler, which translate the entire source program in to machine code.
* the advantage of interpreter compared to compiler is its fast response to change in source program do not require large memory in computer.
* the disadvantage of interpreter is that it is time consuming method because each time a statement in a program is executed then it is first translated.
* thus compiled machine language program runs much faster than an interpreted program.

THAE END