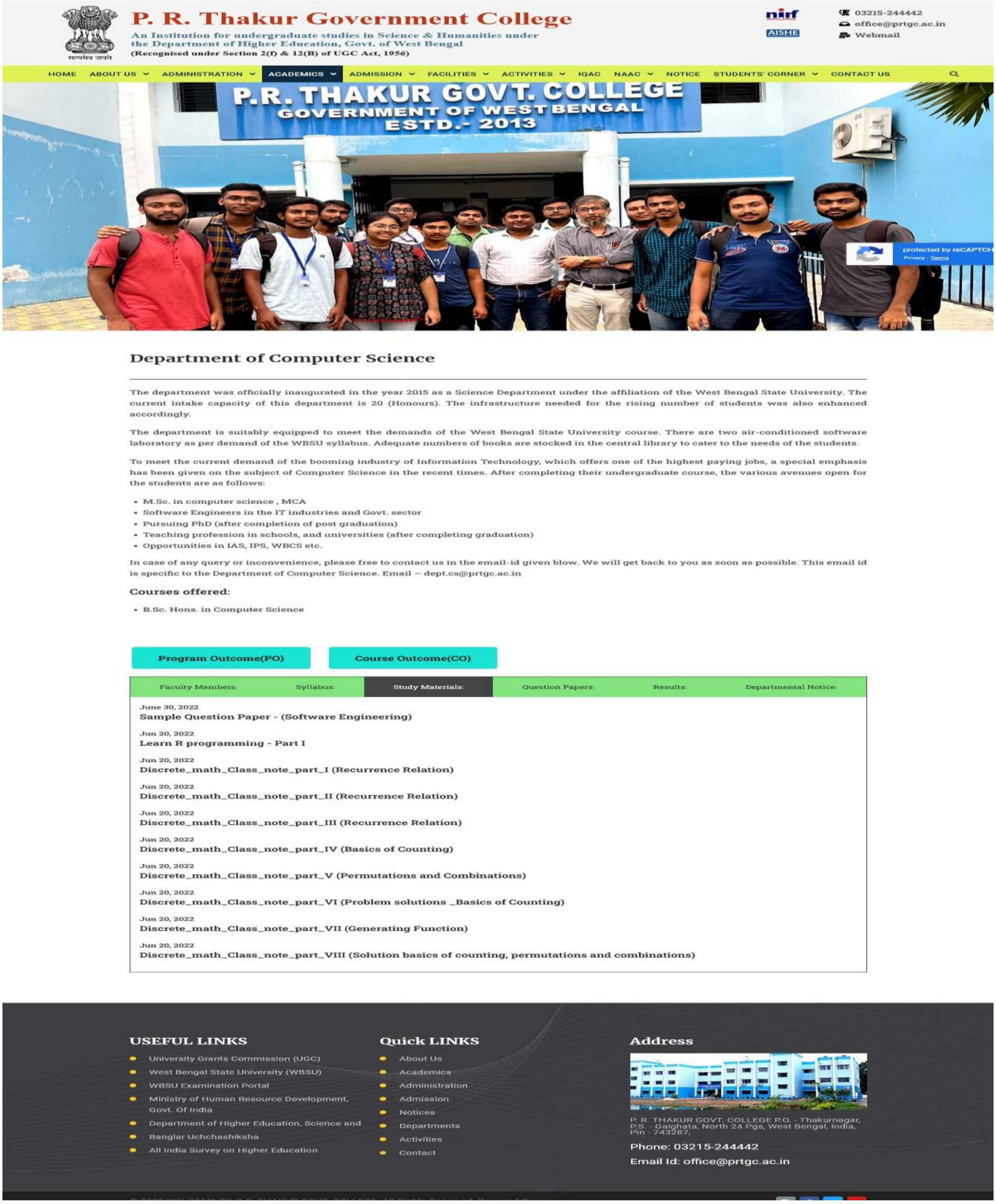
सत्यमेव जयते

OFFICE OF THE PRINCIPAL

P. R. Thakur Government College

P.O.-Thakurnagar, P.S.-Gaighata, North 24 Pgs., Pin-743287
Ph. No. : 913215-244442
Estd. 2013 Email: office@prtgc.ac.in Website: www.prtgc.ac.in

Memo. No.- Date:



P. R. Thakur Government College
An Institution for undergraduate studies in Science & Humanities under the Department of Higher Education, Govt. of West Bengal
(Recognised under Section 2(f) & 12(B) of UGC Act, 1956)

nirf AISHE

☎ 03215-244442
office@prtgc.ac.in
Webmail

HOME | ABOUT US | ADMINISTRATION | ACADEMICS | ADMISSION | FACILITIES | ACTIVITIES | IQAC | NAAC | NOTICE | STUDENTS' CORNER | CONTACT US

P.R. THAKUR GOVT. COLLEGE
GOVERNMENT OF WEST BENGAL
ESTD.- 2013

Department of Computer Science

The department was officially inaugurated in the year 2015 as a Science Department under the affiliation of the West Bengal State University. The current intake capacity of this department is 20 (Honours). The infrastructure needed for the rising number of students was also enhanced accordingly.

The department is suitably equipped to meet the demands of the West Bengal State University course. There are two air-conditioned software laboratory as per demand of the WBSU syllabus. Adequate numbers of books are stocked in the central library to cater to the needs of the students.

To meet the current demand of the booming industry of Information Technology, which offers one of the highest paying jobs, a special emphasis has been given on the subject of Computer Science in the recent times. After completing their undergraduate course, the various avenues open for the students are as follows:

- M.Sc. in computer science , MCA
- Software Engineers in the IT industries and Govt. sector
- Pursuing PhD (after completion of post graduation)
- Teaching profession in schools, and universities (after completing graduation)
- Opportunities in IAS, IPS, WBCS etc.

In case of any query or inconvenience, please free to contact us in the email-id given blow. We will get back to you as soon as possible. This email id is specific to the Department of Computer Science. Email – dept.cs@prtgc.ac.in

Courses offered:

- B.Sc. Hons. in Computer Science

Program Outcome(PO) | Course Outcome(CO)

Faculty Members: | Syllabus: | Study Materials: | Question Papers: | Results: | Departmental Notice:

June 30, 2022
Sample Question Paper - (Software Engineering)

Jun 20, 2022
Learn R programming - Part I

Jun 20, 2022
Discrete_math_Class_note_part_I (Recurrence Relation)

Jun 20, 2022
Discrete_math_Class_note_part_II (Recurrence Relation)

Jun 20, 2022
Discrete_math_Class_note_part_III (Recurrence Relation)

Jun 20, 2022
Discrete_math_Class_note_part_IV (Basics of Counting)

Jun 20, 2022
Discrete_math_Class_note_part_V (Permutations and Combinations)

Jun 20, 2022
Discrete_math_Class_note_part_VI (Problem solutions _Basics of Counting)

Jun 20, 2022
Discrete_math_Class_note_part_VII (Generating Function)

Jun 20, 2022
Discrete_math_Class_note_part_VIII (Solution basics of counting, permutations and combinations)

USEFUL LINKS

- University Grants Commission (UGC)
- West Bengal State University (WBSU)
- WBSU Examination Portal
- Ministry of Human Resource Development, Govt. Of India
- Department of Higher Education, Science and
- Banglar Uchchashiksha
- All India Survey on Higher Education

Quick LINKS

- About Us
- Academics
- Administration
- Admission
- Notices
- Departments
- Activities
- Contact

Address

P. R. THAKUR GOVT. COLLEGE P.O. - Thakurnagar, P.S. - Gaighata, North 24 Pgs, West Bengal, India, Pin - 743287,

Phone: 03215-244442

Email Id: office@prtgc.ac.in

Swapan Sarkar
Officer-in-Charge
P.R. Thakur Govt. College
P.O.-Thakurnagar, 24Pgs.(N)
Pin-743287

Sample Questions
Software Engineering

1. What is the purpose of SRS document?
2. What do you mean by mutation testing?
3. What do you understand by the control structure of a program?
4. Why data dictionary is used in a DFD model of a system?
5. Which are the major phases in the waterfall model of software development? Which phase consumes the maximum effort for developing a typical software product?
6. How are the risk associated with a project handled in the spiral model of software development?
7. Describe the motivation and approaches for integration testing in brief.
8. What is context diagram in DFD? Why is it called level 0 DFD?
9. How path coverage testing method is carried out using control flow graph? Explain with proper example.
10. Which is a stronger testing: data flow testing or path testing? Justify.
11. What are various types of software maintenance technique?
12. What is recursive relationship in an ER model?
13. What are the advantades and disadvantages of using iterative waterfall model?
14. Explain with an example how a DFD model can be transformed into structure chart.
15. Explain twp software quality matrics: Robustness and Reusability.
16. Write brief about the role of system analyst.
17. What do you mean by V-model in SDLC?
18. What is coghesion and coupling?
19. Distinguish between software verification and software validation?
20. Discuss spiral model for SDLCs? Why it is called meta model?
21. Distiguish between structure chart and flow chart?
22. What does the term "balancing a DFD" mean? Give an example to explain your answer.
23. Consider the following code of segment and construct Control Flow Graph of it. Then use it to find optimal test cases with the help of Path Coverage Criteria during white box testing:

```
func(x,y)
{
1       while(x!=y){
2           if(x>y)
3               x=x-y;
4           else    y=y-x;
5       }
6       return x;
}
```

24. Define software quality.
25. What do you menat by system testing?
26. Draw a schematic diagram to represent Waterfall Model of Software Development.
27. Explain the features of prototyping model. What are its advantage and limitations?
28. Distiguish between a physical and logical DFD. Discuss how a physical DFD can be transformed into a logical DFD.

29. What is software testing? Distinguish between white-box testing and black-box testing.
30. Write a prograam to add two digit integers. Find how many test cases are required to test the program completely?
31. State attributtes of a "Quality Software"
32. Discuss the spiral model for SDLC and mention the utility of it over waterfall model.
33. In a Hospital Managment System, develop a DFD for a "Ward servide management system". State all your assumptions.
34. Differentiate between alpha testing and beta testing.
35. Define Cyclomatic complexity.
36. What is meta data?
37. What do you understand by view level in data model?
38. Why is low coupling desirable?
39. Discuss the importance of analysis and design in software life cycle.
40. Discuss characterstics of Spiral model? Why is Spiral model difficult to implement.
41. Draw level-0 and level- DFD of a Library Management System. Make suitable assumptions.
42. How are the risks associated with a project handled in the spiral model of software development?
43. Lisr four characterstics of a good Software Requirements Specification.
44. What are the main shortcomings of DFD as a tool for performing structured analysis.
45. Explain COCOMO model.

R

Programming
Part - I

File creation:

```
$ gedit first_program.r
```

Write Code:

```
# My first program in R Programming
Strings <- "Hello, World!"
print ( Strings)
```

Run:

```
$ Rscript first_program.r
```

Output:

```
[1] "Hello, World!"
```

R - Data Types

Vectors Lists Matrices Arrays Factors Data Frames

Vectors		

Logical	TRUE, FALSE	Create a vector:
		fruit <- c('mango','banana',"pineapple")
		print(fruit)

Numeric	3.14, 4, 1000	
Integer	2L, 34L, 0L	
Complex	2+3i	
Character	'A' , "Hello", "TRUE", '13.4'	
Raw	"Hello" is stored as 48 65 6c 6c 6f	

Vectors:

```
> fruit <- c('mango','banana',"pineapple")
> print(fruit)
[1] "mango"    "banana"    "pineapple"
>

> element <- c(2,3,4,1,7,8)
> print(element)
[1] 2 3 4 1 7 8
>
```

Lists:

```
> L <- list(c(10,20,30),3.14,sin)

> print(L)

[[1]]

[1] 10 20 30

[[2]]

[1] 3.14

[[3]]

function (x)  .Primitive("sin")
```

Matrix:

```
> M = matrix( c('x','y','z','y','z','x'), nrow = 2, ncol = 3, byrow = TRUE)
> print(M)
     [,1] [,2] [,3]
[1,] "x"  "y"  "z"
[2,] "y"  "z"  "x"
>
```

Array:

```
> a <- array(c(0,1),dim = c(3,3))
> print(a)
     [,1] [,2] [,3]
[1,]    0    1    0
[2,]    1    0    1
[3,]    0    1    0
>



> a <- array(c(0,1),dim = c(3,3,2))
> print(a)
, , 1

     [,1] [,2] [,3]
[1,]    0    1    0
[2,]    1    0    1
[3,]    0    1    0

, , 2

     [,1] [,2] [,3]
[1,]    1    0    1
[2,]    0    1    0
[3,]    1    0    1
>
```

Factors:

```
> fruit<-c('pineapple','mango','banana','mango','pineapple','mango','banana')
> factor_object_fruit <- factor(fruit)
> print(factor_object_fruit)

[1] pineapple mango     banana    mango     pineapple mango     banana
Levels: banana mango pineapple

> print(nlevels(factor_object_fruit))
[1] 3
>
```

Data Frame:

```
# Create the data frame.

Record <-       data.frame(

 Name = c("Avik", "Ayan","Sukanta"),

   Roll = c(3, 1, 4),

   Stream = c('BA','BSC','BA'),

   Department = c('Bengali','Math','English')

)

print(Record)

     Name Roll Stream Department

1    Avik    3     BA    Bengali

2    Ayan    1    BSC       Math

3 Sukanta    4     BA    English

>
```

If-else statement:

if (test_expression) {
statement
}
	x <- 5
	if(x > 0){
	print("Positive number")
	}
	 else {
	        print("Negative number")
	        }

Loop statement:

Repeat:

repeat {
statement
}
	x <- 1
	repeat {
	print(x)
	x = x+1
	if (x == 6){
	break
	}
	}

While:

while (test_expression)
{
statement
}
	i <- 1
	while (i < 10) {
	print(i)
	i = i+1
	}

For:

for (val in sequence)
{
statement
}
	i=1
	for (i in 1:10) {
	  i=i+1
	  }
	  print(i)

Functions:

Code:

```
sum <- function(a) {
  sum <- 0
  for(i in 1:a) {
    sum=sum+i

  }
  print(paste("sum=",sum))
}
cat("Enter the value of n: ");
a<-readLines("stdin",n=1);
sum(a)
```

Output:

```
Enter the value of n: 5
[1] "sum= 15"
```

1. Write a program to check a number prime of not.

Program Code

```
# Program to check if the input number is prime or not
# take input from the user
cat("Enter a positive number: ");
num <- readLines("stdin",n=1);
num<-as.integer(num)
flag = 0
# prime numbers are greater than 1
if(num > 1) {
# check for factors
flag = 1
for(i in 2:(num-1)) {
if ((num %% i) == 0) {
flag = 0
break
}
}
}
if(num == 2)    flag = 1
if(flag == 1) {
print(paste(num,"is a prime number"))
} else {
print(paste(num,"is not a prime number"))
}
```

```
probir@Incredible:~/Documents/R_program$ Rscript prime_number.R
Enter a positive number: 13
[1] "13 is a prime number"
```

2. Write a program to check the given number arstrong number of not.

Program: Code:

```
# take input from the user
cat("Enter a positive number: ");
num <- readLines("stdin",n=1);
num<-as.integer(num)
count=0
# initialize sum
sum = 0

# find the sum of the cube of each digit
temp = num
while(temp > 0) {
temp = floor(temp / 10)
count=count+1
}
temp=num
```

```
while(temp > 0) {
digit = temp %% 10
sum = sum + (digit ^ count)
temp = floor(temp / 10)
}
# display the result
if(num == sum) {
print(paste(num, "is an Armstrong number"))
} else {
print(paste(num, "is not an Armstrong number"))
}
```

Output:

```
probir@Incredible:~/Documents/R_program$ Rscript amstrong.R
Enter a positive number: 370
[1] "370 is an Armstrong number"
```

3. Write a program to find the factorial of a given number.

Program: Code:

```
facto <- function(){
# accept the input provided by the user and convert to integer
cat("Enter a positive number: ");
num <- readLines("stdin",n=1);
num<-as.integer(num)

fact = 1
# checking whether the number is negative, zero or positive
if(num < 0) {
print(" The number is negative the factorial does not exist. ")
} else if(num == 0) {
print(" The factorial result is 1 ")
} else {
for( i in 1:num) {
fact = fact * i
}
print(paste(" The factorial result is ", num ,"is", fact ))
}
}
facto()
```

Output:

```
Enter a positive number: 5
[1] " The factorial result is  5 is 120"
```

4.Write a program to sort an array in ascending order.

Program Code:

```
#Bubble Sort
sort.b <- function(x)
{
  if(!is.unsorted(x)) {stop("Vector is already sorted")}
  if(length(x)<2){stop("vector is not long enough") }
  if ( !is.vector(x) )  { stop("parameter must be a vector") }
  if ( !is.numeric(x) ) { stop("parameter must be numeric") }

  n = length(x)
  v = x

  for(j in 1:(n-1))
  {
    for(i in 1:(n-j))
    {
      if(v[i+1]<v[i])
      {
        t = v[i+1]
        v[i+1] = v[i]
        v[i] = t
      }
    }
  }
  print(v)
  x = v
}
cat("Enter number of elements: ");
n <- readLines("stdin",n=1);
n<-as.numeric(n)
cat("Enter elements: ");
x<-c()
for(i in 1:n)
{
x[i] <- readLines("stdin",n=1);

i=i+1
}
x<-as.numeric(x)
sort.b(x)
```

Output:

```
Enter number of elements: 5
Enter elements: 3
2
-1
5
1
[1] -1  1  2  3  5
```